

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



SISTEMA DE LOCALIZACIÓN EN REDES Wi-Fi
BASADO EN FINGERPRINT

PROYECTO FIN DE CARRERA
INGENIERA TÉCNICA: SISTEMAS DE
TELECOMUNICACIÓN

Autora: M^a Ángeles Núñez Sobrino
Tutor: Julio Villena Román

Título: Sistema de localización en redes Wi-Fi basado en Fingerprint.

Autor: M^a Ángeles Núñez Sobrino

Tutor: Julio Villena Román

EL TRIBUNAL

Presidente:

Secretario:

Vocal:

Realizado el acto de defensa del Proyecto Fin de Carrera el día de de 200 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

Fdo: Presidente

Fdo: Vocal

RESUMEN

La gran difusión de los sistemas de localización *outdoor*, y en concreto GPS, que permiten obtener la localización geográfica en un mapa de un determinado dispositivo, han provocado el desarrollo de una demanda de funcionalidades similares a esta tecnología pero válida en el interior de edificios. En esta línea, el presente proyecto aprovecha la tecnología Wi-Fi para desarrollar un sistema de localización cuya funcionalidad sea ubicar a un dispositivo móvil conectado a una red basada en el estándar 802.11.

La localización del dispositivo móvil se basa en determinar la posición del punto de acceso (AP) de la red que se encuentre más cerca de éste. Cada punto de acceso se encuentra registrado en una base de datos, en concreto, la información que identifica cada dispositivo, la potencia de la señal recibida (RSSI). Esta es la técnica conocida como *Fingerprint*, que consiste en la comparación de las señales recibidas en la posición actual que se quiere determinar con las mediciones de las señales recibidas obtenidas previamente de la zona cubierta por el sistema, durante la fase denominada *off-line* o de entrenamiento.

La otra fase del proceso, la que se conoce como *real-time*, consiste en la localización propiamente dicha del dispositivo, para lo cual se calcula la distancia entre dicho dispositivo y los puntos de acceso registrados en la fase anterior, utilizando para la clasificación el algoritmo k-NN. La distancia más corta, corresponderá al dispositivo que se encuentra más próximo al dispositivo a localizar, y como consecuencia, éste adoptará la ubicación del dispositivo más cercano.

Además, se ha desarrollado como caso de uso la implantación del sistema de localización en las instalaciones del Departamento de Telemática de la Universidad Carlos III de Madrid. En este emplazamiento se han realizado una serie de pruebas con el sistema para intentar determinar los parámetros óptimos.

INDICE

1.- INTRODUCCIÓN.....	1
1.1 MOTIVACIÓN	1
1.2 OBJETIVOS	2
1.2.1 Parámetros.....	2
1.2.2 Atributos.....	2
1.2.3 Medida de diferenciación	2
1.2.4 Modificaciones en el procedimiento.....	2
1.3 ESTRUCTURA DEL PROYECTO	3
2.- ESTADO DEL ARTE.....	4
2.1 SISTEMAS DE LOCALIZACIÓN.....	4
2.1.1 Definición.....	4
2.1.2 Clasificación de los sistemas de localización.	4
2.2 TECNOLOGÍAS INALÁMBRICAS.....	6
2.2.1 Infrarrojos (IR)	6
2.2.1.1 Definición	6
2.2.1.2 Historia.....	6
2.2.1.3 IrDA (Infrared Data Association).....	7
2.2.2 Ultrasonido	7
2.2.2.1 Definición	7
2.2.2.2 Historia.....	7
2.2.2.3 Funcionamiento.....	7
2.2.3 RFID.....	8
2.2.3.1 Definición	8
2.2.3.2 Historia.....	8
2.2.3.3 Funcionamiento.....	8
2.2.4 Ultra Wide Band (UWB).....	9
2.2.4.1 Definición	9
2.2.4.2 Historia.....	9
2.2.4.3 Funcionamiento.....	10
2.2.5 Bluetooth	10
2.2.5.1 Definición	10
2.2.5.2 Historia.....	10
2.2.5.3 Funcionamiento.....	11
2.2.6 WLAN.....	12
2.2.6.1 Definición	12
2.2.6.2 Historia.....	12
2.2.6.3 Arquitectura 802.11	13
2.2.6.4 Estándar IEEE 802.11	16
2.2.6.6 Otras tecnologías que compiten con IEEE 802.11	17
2.2.6.7 Comparativa entre estándares 802.11x y tecnologías competidoras.....	18
2.2.7 WiMAX.....	18
2.2.8 ZigBee	19
2.3 MÉTRICAS DE LA SEÑAL.....	19
2.3.1 AOA (Angle Of Arrival)	19

2.3.1.1	Definición	19
2.3.1.2	Método clásico de estimación de AOA	19
2.3.1.3	Errores de la estimación de la dirección.....	20
2.3.1.4	Exactitud de la técnica	20
2.3.2	TOA (Time of arrival)	21
2.3.2.1	Definición	21
2.3.2.2	Método estimación de TOA	21
2.3.2.3	Errores de la estimación de la dirección.....	22
2.3.2.4	Exactitud de la técnica	23
2.3.3	TDOA (Time difference of arrival)	23
2.3.3.1	Definición	23
2.3.3.2	Método estimación TDOA	23
2.3.3.3	Exactitud de la técnica	23
2.3.4	RSSI (Receive Signal Strength Indication)	23
2.3.4.1	Definición	23
2.3.4.2	Método de estimación	23
2.3.4.3	Errores de la estimación de la dirección.....	24
2.3.4.4	Precisión de la técnica.....	24
2.3.5	Cell- Id.....	24
2.3.6	Comparación entre TOA, AOA, RSSI y Cell-Id	24
2.5	CLASIFICACIÓN DE LOS SISTEMAS DE LOCALIZACIÓN	25
2.5.1	Triangulación.....	25
2.5.1.1	Definición	25
2.5.1.2	Técnica de triangulación	25
2.5.2	Técnica Fingerprint	27
2.5.2.1	Definición de Fingerprint	27
2.5.2.2	Metodología de la técnica Fingerprint.....	27
2.5.2.3	Predicción Fingerprint.....	27
2.6	ALGORITMOS DE UN SISTEMA DE LOCALIZACIÓN	28
2.6.1	Métodos heurísticos: Teoría Bayesiana	29
2.6.1.1	Definición	29
2.6.1.2	Técnicas de localización Bayesiana	30
2.6.1.3	Problemas.....	30
2.6.2	Métodos heurísticos: Redes neuronales.....	30
2.6.2.1	Definición	30
2.6.2.3	Estructura de la neurona.....	31
2.6.2.4	Arquitectura de red.....	31
2.6.2.5	Algoritmos de aprendizaje (entrenamiento)	32
2.6.3	Métodos heurísticos: k vecinos más cercanos	34
2.6.3.1	Definición	34
2.6.3.2	Algoritmo KNN	34
2.6.3.3	Ventajas	35
2.6.3.4	Desventajas	35
2.6.3.5	Técnicas de reducción del conjunto de entrenamiento a aplicar en algoritmos basados en aprendizaje de ejemplos	35
3.-	DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	39
3.1	ELECCIÓN DEL DISEÑO DEL SISTEMA	39
3.1.1	Bloques del sistema de localización	39
3.1.1.1	Estimación de los parámetros de error	39
3.1.1.2	Estimación de la localización de dispositivos	39

3.1.2 Fases de localización	39
3.1.2.1 Toma de muestras	40
3.1.2.2 Fase Off-Line o de Training.....	41
3.1.2.3 Fase Real-Time	42
3.1.3 Técnicas de optimización del sistema de localización.....	42
3.1.3.1 Técnicas de optimización en la fase Real –Time.....	42
3.1.3.2 Técnicas de optimización en la fase Off –Line:	45
3.2 IMPLEMENTACIÓN DEL SISTEMA.....	46
3.2.1 Conjunto de entrenamiento y de testing	46
3.2.2 Construcción de los Fingerprint	46
3.2.3 Cálculo de distancias	47
3.2.4 Decisión.....	47
3.2.5 Otras clases adicionales	47
4.- EVALUACIÓN DEL SISTEMA	48
4.1 FASE I: SISTEMA DE LOCALIZACIÓN CON 80% DE INSTANCIAS DE ENTRENAMIENTO	49
4.1.1 K y N variables.....	49
4.1.2 Pesos.....	50
4.1.3 Distancias	51
4.1.4 Normalización	52
4.1.4.1 Comparativa del comportamiento del algoritmo sin y con normalización de datos para $k \in [2,8]$	52
4.1.4.2 Comportamiento del algoritmo con la propiedad del peso y con los datos normalizados.....	53
4.1.4.3 Comparativa de la importancia de la presencia del peso en el caso de los datos sin normalizar y normalizados.....	55
4.1.4.4 Comparativa entre las diferentes distancias sin peso y con normalización.....	57
4.1.4.5 Comparativa entre las diferentes distancias sin la propiedad del peso, normalizando y sin normalizar los datos	57
4.1.4.6 Comparativa entre diferentes distancias con la propiedad del peso y normalizando	58
4.1.4.7 Comparativa entre las diferentes distancias normalizadas con y sin peso para $k \in [2, 8]$	58
4.1.4.8 Comparativa entre las diferentes distancias con la propiedad del peso, normalizando y sin normalizar para $k \in [2,8]$	60
4.1.5 Estudio para las diferentes distancias	61
4.1.5.1 Resultados obtenidos con la Distancia Euclídea	61
4.1.5.2 Resultados obtenidos con la distancia City Block.....	61
4.1.5.3 Resultados obtenidos con la distancia Chebichef.....	62
4.1.5.4 Resultados obtenidos con la distancia Canberra.....	63
4.2 FASE II: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training.....	64
4.2.1 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training N variable, sin peso y sin normalización para $k \in [2,8]$	64
4.2.2 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento N variable, sin peso y con normalización para $k \in [2,7]$	66
4.2.3 Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento N variable, con peso y sin normalización para $k \in [2,7]$	67
4.2.4 Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento N variable, con peso y con normalización para $k \in [2,7]$	69
4.2.5 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y sin normalización para $k \in [2,8]$	70

4.2.6 Estudio de las diferentes distancias	71
4.2.7 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y con normalización para $k \in [2,7]$	72
4.2.8 Comparativa del comportamiento del algoritmo con diferentes tamaños de conjunto de entrenamiento y distancias, con peso y sin normalización para $k \in [2,7]$	73
4.2.9 Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de training y distancias, con peso y con normalización para $k \in [2,8]$	75
4.3 FASE III: Optimización del algoritmo	76
4.3.1 Carga computacional	76
4.3.2 Filtrado de ruido	78
4.3.2.1 Comparativa entre KNN y ENN con k variable, sin peso y sin normalización de datos ..	78
4.3.2.2 Comparativa entre KNN y ENN con k variable, con peso y sin normalización de datos ..	79
4.3.2.3 Comparativa entre KNN y ENN con k variable, sin peso y con normalización de datos ..	79
4.3.2.4 Comparativa entre KNN y ENN con k variable, con peso y con normalización de datos ..	80
4.4 FASE IV: Resumen de resultados	81
5.- CONCLUSIONES Y TRABAJOS FUTUROS.....	82
5.1 PARÁMETROS.....	82
5.2 ATRIBUTOS	82
5.3 MEDIDAS DE DIFERENCIACIÓN	82
5.4 MODIFICACIONES DEL PROCEDIMIENTO.....	83
5.4.1 K-dd dinámico	83
5.4.2 ENN.....	83
5.5 TRABAJOS FUTUROS	83
5.5.1 Integración de un sistema de localización completo	83
5.5.2 Optimización del código.....	83
5.5.3 k-NN aplicado a objetos en movimiento	83
5.5.4 Localización en 3 Dimensiones	84
ANEXO	85
FASE I	85
FASE II: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento	94
FASE III: Optimización del algoritmo	102
BIBLIOGRAFÍA	107

ÍNDICE DE FIGURAS

Figura 1: Diagrama de bloques de un Sistema de Localización.	4
Figura 2: Comunicación punto a punto del estándar IrDA.	7
Figura 3: Niveles del estándar 802.11	13
Figura 4: Independent Basic Service Set (IBSS)	13
Figura 5: Extended Service Set (ESS) – multiple cells	14
Figura 6: Modelo de estimación de AOA.	20
Figura 7: Método de triangulación TOA.	22
Figura 8: Método de triangulación TDOA.	23
Figura 9: Técnica de lateración.	26
Figura 10: Técnica de angulación.	26
Figura 11: Modelo de Mc Culloch- Pitts	31
Figura 12: Red neuronal FeedForward.	32
Figura 13: Red neuronal recurrente.	32
Figura 14: Plano del entorno del sistema de localización.	40
Figura 15: Volcado de los datos de training.	41
Figura 16: Estudio del comportamiento del algoritmo variando los parámetros k y N.	49
Figura 17: Estudio del comportamiento del algoritmo variando los parámetros k y N.	50
Figura 18: Comparativa del comportamiento del algoritmo sin la propiedad del peso y con ella para $k \in [2,8]$	51
Figura 19: Evaluación de diferentes distancias sin $k=1$, sin normalizar y $N=68$	51
Figura 20: Comparativa entre diferentes distancias con la propiedad el peso y sin normalizar.	52
Figura 21: Comparativa del comportamiento del algoritmo sin y con normalización de datos para.....	53
Figura 22: Comportamiento del algoritmo con la propiedad del peso y con los datos normalizados.....	54
Figura 23: Comparación del comportamiento del algoritmo usando la propiedad del peso y con y sin normalización para $k \in [2, 8]$	55
Figura 24: Comparativa entre las diferentes distancias sin peso y con normalización.	57
Figura 25: Comparativa entre las diferentes distancias sin la propiedad del peso, normalizando y sin normalizar para $k \in [2, 8]$	58
Figura 26: Comparativa entre las diferentes distancias con la propiedad del peso y normalizando.	58
Figura 27: Comparativa entre las diferentes distancias normalizadas con y sin peso para $k \in [2, 8]$	59
Figura 28: Comparativa entre las diferentes distancias con la propiedad del peso, normalizando y sin normalizar para $k \in [2, 8]$	60
Figura 29: Resultados obtenidos con la Distancia Euclídea.	61
Figura 30: Resultados obtenidos con la Distancia Euclídea.	61
Figura 31: Resultados obtenidos con la Distancia City Block.....	62
Figura 32: Resultados obtenidos con la Distancia City Block.....	62
Figura 33: Resultados obtenidos con la Distancia Chevichef.....	63
Figura 34: Resultados obtenidos con la Distancia Chevichef.....	63
Figura 35: Resultados obtenidos con la Distancia Canberra.	64

Figura 36: Resultados obtenidos con la Distancia Canberra.	64
Figura 37: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, sin peso y sin normalización para todo $k \in [2, 7]$	65
Figura 38: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de training, N variable, sin peso y sin normalización para todo $k \in [2, 7]$	66
Figura 39: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, sin peso y con normalización para todo k para todo $k \in [2, 7]$	67
Figura 40: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupos de training, N variable, sin peso y con normalización para todo k para todo $k \in [2, 7]$	67
Figura 41: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y sin normalización para todo $k \in [2, 7]$	68
Figura 42: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training, N variable, con peso y sin normalización para k	68
Figura 43: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y con normalización para $k \in [2, 7]$	69
Figura 44: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y con normalización para $k \in [2, 7]$	70
Figura 45: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y sin normalización para $k \in [2, 7]$	71
Figura 46: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training y distancias, sin peso y sin normalización para $k \in [2, 7]$	72
Figura 47: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y con normalización para $k \in [2, 7]$	72
Figura 48: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de entrenamiento y distancias, sin peso y con normalización para $k \in [2, 7]$	73
Figura 49: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de training y distancias, con peso y sin normalización para $k \in [2, 7]$	74
Figura 50: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, con peso y sin normalización para $k \in [2, 7]$	74
Figura 51: Comparativa del comportamiento del algoritmo con diferentes tamaños de conjuntos de entrenamiento y distancias, con peso y con normalización para $k \in [2, 7]$	75
Figura 52: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de entrenamiento y distancias, con peso y con normalización para $k \in [2, 7]$	76
Figura 53: Comparativa entre KNN y ENN con k variable, sin peso y sin normalización de datos.	79
Figura 54: Comparativa entre KNN y ENN con k variable, con peso y sin normalización de datos.	79
Figura 55: Comparativa entre KNN y ENN con k variable, sin peso y con normalización de datos.	80
Figura 56: Comparativa entre KNN y ENN con k variable, con peso y con normalización de datos.	81

ÍNDICE DE TABLAS

Tabla 1: Comparación de técnicas de localización.....	24
Tabla 2: Comparativa de efectividad de la presencia del peso en los datos normalizados y sin normalizar.	55
Tabla 3: Resultado del algoritmo de localización k-NN básico y k-NN dinámico.....	77
Tabla 4: Resultados óptimos.	83
Tabla 5: Estudio del comportamiento del algoritmo variando los parámetros k y N.....	85
Tabla 6: Comparativa del comportamiento del algoritmo sin la propiedad del peso y con ella para $k \in [2,8]$	85
Tabla 7: Evaluación de diferentes distancias sin peso, sin normalizar y $N=68$	86
Tabla 8: Comparativa entre diferentes distancias con la propiedad el peso y sin normalizar.....	86
Tabla 9: Comparativa del comportamiento del algoritmo sin y con normalización de datos para $k \in [2,8]$	87
Tabla 10: Comportamiento del algoritmo con la propiedad del peso y con los datos normalizados.....	87
Tabla 11: Comparación del comportamiento del algoritmo usando la propiedad del peso y con y sin normalización para $k \in [2, 8]$	88
Tabla 12: Comparativa de la importancia de la presencia del peso en el caso de los datos sin normalizar y normaliza.....	89
Tabla 13: Comparativa entre las diferentes distancias sin peso y con normalización	90
Tabla 14: Comparativa entre las diferentes distancias sin la propiedad del peso, normalizando y sin normalizar los datos.....	90
Tabla 15: Comparativa entre diferentes distancias con la propiedad del peso y normalizando.....	91
Tabla 16: Comparativa entre las diferentes distancias normalizadas con y sin peso para $k \in [2,8]$	91
Tabla 17: Comparativa entre las diferentes distancias con la propiedad del peso, normalizando y sin normalizar para $k \in [2,8]$	92
Tabla 18: Resultados obtenidos con la Distancia Euclídea	92
Tabla 19: Resultados obtenidos con la distancia City Block.....	93
Tabla 20: Resultados obtenidos con la distancia Chevichef.	93
Tabla 21: Resultados obtenidos con la distancia Canberra.....	93
Tabla 22: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, sin peso y sin normalización para $k \in [2, 7]$	94
Tabla 23: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de training, N variable, sin peso y con normalización para $k \in [2,7]$	95
Tabla 24: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, con peso y sin normalización para $k \in [2,7]$	96

Tabla 25: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y con normalización para $k \in [2,7]$	97
Tabla 26: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training y distancias, sin peso y sin normalización para $k \in [2,8]$	98
Tabla 27: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento y distancias, sin peso y con normalización para $k \in [2,7]$	99
Tabla 28: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training y distancias, con peso y sin normalización para $k \in [2,7]$	100
Tabla 29: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, con peso y con normalización para $k \in [2,7]$	101
Tabla 30: Resultado del algoritmo de localización k-NN básico y k-NN dinámico	102
Tabla 31: Comparativa entre KNN y ENN con k variable, sin peso y sin normalización de datos	103
Tabla 32: Comparativa entre KNN y ENN con k variable, con peso y sin normalización de datos	104
Tabla 33: Comparativa entre KNN y ENN con k variable, sin peso y con normalización de datos	105
Tabla 34: Comparativa entre KNN y ENN con k variable, con peso y con normalización de datos	106

1.- INTRODUCCIÓN

1.1 MOTIVACIÓN

En la sociedad actual, existe una demanda creciente de las llamadas aplicaciones basadas en localización (*Location Based Application*), que podrían definirse como aquellos sistemas que buscan ofrecer un servicio personalizado a los usuarios basado en información de su ubicación geográfica. Algunos ejemplos típicos son servicios de mapas, enrutamiento (ayuda a la navegación) y páginas amarillas geográficas.

Para ello existen diferentes tecnologías, incluyendo sistemas de información geográfica, tecnologías de posicionamiento (GPS, servicio de posicionamiento suministrado por el operador de red) y tecnología de comunicaciones para transmitir información a la aplicación.

La posibilidad de localizar y ser localizado no sólo se demanda en los entornos *outdoor* (al aire libre), sino también dentro de áreas cerradas, entorno conocido como *indoor* (bajo techo). Por ejemplo, en hospitales, donde conocer la ubicación del personal médico en el menor tiempo posible, aumentaría la eficiencia del servicio sanitario. Los centros comerciales, museos y aeropuertos son lugares en los que serían muy útiles mapas para conocer la ubicación de los sitios de interés. La posibilidad de poder descargar un plano en un dispositivo, compatible con la tecnología existente, permitiría al usuario optimizar el tiempo durante su visita en estos emplazamientos. Si junto al mapa se añade la posibilidad de un guía virtual que indicara los lugares de interés y la posición que se encuentra el dispositivo con respecto a un punto, permitiría que las discapacidades visuales de algunas personas, no fueran ningún inconveniente a la hora de familiarizarse con estos edificios.

Hoy en día la tecnología más extendida es la tecnología GPS, basada en localización mediante triangulación de señales emitidas por una constelación de satélites, pero sólo es válida en localizaciones *outdoor*. En entornos cerrados, debido a sus diferentes características, no es posible utilizar GPS porque la señal de los satélites no llega con suficiente potencia. En estos entornos se aplican otras técnicas, como UWB, Bluetooth, WLAN, que pretenden conseguir el mismo objetivo.

Entre estas tecnologías, este proyecto se centrará en las redes WLAN, conocidas comúnmente como Wi-Fi, implantadas en el interior de entornos cerrados mediante el estándar 802.11 del IEEE. La facilidad de su instalación, ya que se despliega sin necesidad de cableado, y el bajo coste, gracias a la competencia entre los vendedores que ha provocado el descenso drástico de su precio en los últimos años, hacen que la presencia de la tecnología Wi-Fi sea cada vez más extendida en todo tipo de edificios. A estas ventajas también se puede añadir la interoperabilidad, gracias al empleo de estándares, con los que pueden combinarse dispositivos de fabricantes diferentes. Además, las capacidades de roaming entre los diferentes dispositivos que forman una red inalámbrica es un punto más a su favor.

Sin embargo, la tecnología WLAN tiene una característica desfavorable, que consiste en que trabaja en la banda de 2.4 GHz. En este rango de frecuencias, hay problemas que pueden interferir, degradar la calidad de la comunicación y resultar vulnerable y poco segura. Gracias a las ventajas anteriores y a la demanda del público, un gran número de empresas siguen centrando sus investigaciones en mejorar sus puntos débiles y potenciar los fuertes.

Por tanto, se puede concluir que la motivación del trabajo aquí presentado es el mencionado auge en la demanda de aplicaciones basadas en localización, y se centra en proporcionar una solución para la localización de dispositivos móviles en el interior de emplazamientos cerrados, basándose en la tecnología WLAN, y, en concreto, como se verá posteriormente, en la técnica de mapa radio o *Fingerprint*.

1.2 OBJETIVOS

El objetivo principal del presente proyecto es construir un sistema de localización óptimo de dispositivos móviles para un entorno *indoor* basado en la tecnología WLAN. Para determinar la posición de un dispositivo, se utiliza como fuente de información la potencia (RSSI) que emiten los dispositivos de red denominados Puntos de Acceso (AP).

Entre las diferentes técnicas existentes, el sistema se basará en la técnica de mapa radio, más comúnmente llamada *Fingerprint*, por su simplicidad y buenos resultados. Esta técnica, como se detallará más adelante, consiste en la comparación de las señales recibidas en la posición actual que se quiere determinar con las mediciones de las señales recibidas obtenidas previamente de la zona cubierta por el sistema, durante la fase denominada *off-line* o de entrenamiento. Como algoritmo de clasificación se utilizará el algoritmo de los k-Vecinos Más Cercanos (k-Nearest Neighbour) o k-NN, de nuevo por su simplicidad y su baja tasa de error. A lo largo de la presente memoria se justificará la elección de ambas tecnologías.

Además, como prueba de concepto, el sistema se validará aplicándolo a la determinación de la posición de un dispositivo móvil en las instalaciones del Departamento de Ingeniería Telemática, en la segunda planta del edificio Torres Quevedo del Campus de Leganés de la Universidad Carlos III de Madrid.

El diseño del sistema de localización viene definido por parámetros, atributos, técnicas, medidas de diferenciación y diferentes variaciones del algoritmo. El objetivo de diseño es minimizar el error de determinación de la localización del dispositivo móvil. Por ello, desde un punto de vista técnico, la intención de este proyecto se puede expresar en los objetivos que se presentan a continuación, que están relacionados, con los elementos que influyen en el funcionamiento del sistema de localización

1.2.1 Parámetros

Obtener el valor óptimo de los parámetros que se detallan a continuación:

- **N:** número de señales de potencia recibidas a considerar que forman el vector de muestras.
- **k:** número de vecinos más cercanos que determinan la decisión del algoritmo de localización.
- **Tamaño del conjunto de entrenamiento:** número de instancias que determinan una nueva ubicación.

1.2.2 Atributos

- **Pesos:** Determinar si tener en cuenta el orden de los resultados obtenidos en el algoritmo k-NN influye negativa o positivamente en los resultados del decisor global.
- **Normalización:** Averiguar si aplicar alguna función de suavizado al valor de las potencias o diferencia de muestras RSSI ayuda a mejorar el número de localizaciones exitosas.

1.2.3 Medida de diferenciación

- Definir cuál de los diferentes métodos posibles para calcular la distancia entre dos instancias es adecuado para el experimento que se trata, es decir, determinarla métrica o medida de distancia más adecuada

1.2.4 Modificaciones en el procedimiento

- Limitar el número de instancias a comparar con una nueva entrada.
- Eliminar los vectores que sean producto del efecto *multipath* y/o desvanecimiento de la señal WLAN que influyen negativamente en la determinación de una nueva localización.

1.3 ESTRUCTURA DEL PROYECTO

En este apartado se pretende ofrecer una panorámica del contenido de cada uno de los capítulos que forman el presente proyecto.

- Capítulo 1: Introducción

El presente capítulo.

- Capítulo 2: Estado del arte

El contenido del capítulo se divide en diferentes apartados que presentan cada uno de los bloques que intervienen en un sistema de localización.

Los puntos a destacar son:

- **Definición de un sistema de localización.**
- **Tecnologías de localización.**
- **Métricas de la señal.**
- **Interferencias: *Multipath***
- **Algoritmos de localización.**

- Capítulo 3: Diseño e implementación

En este apartado se detalla cómo se ha realizado el diseño y la implementación del sistema basado en la técnica *Fingerprint* y el algoritmo k-NN.

- Capítulo 4: Evaluación del sistema

Se detalla el caso de uso donde se ha validado el sistema de localización propuesto, presentando exhaustivamente los resultados de las diferentes pruebas realizadas con el algoritmo utilizado dentro del entorno que engloba el sistema de localización elegido.

- Capítulo 5: Conclusiones y Trabajos Futuros

Finalmente se presentan las conclusiones del trabajo realizado, junto con posibles trabajos futuros continuación del presente proyecto.

- Anexo

Incluye figuras y tablas adicionales que complementan los resultados de la evaluación del sistema.

2.- ESTADO DEL ARTE

2.1 SISTEMAS DE LOCALIZACIÓN

2.1.1 Definición.

Localización es el proceso [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] por el cual se estima la posición de un elemento (dispositivo y/o individuo) dentro de un entorno determinado, mediante cualesquiera técnicas.

Un sistema de localización consiste en:

- Un conjunto de nodos o puntos de acceso (Access Point AP) que emiten una señal al medio (radio, ultrasonidos, luz) y actúan como dispositivos de posicionamiento.
- Una serie de elementos electrónicos que reciben esa misma señal después de propagarse por el medio.
- Una vez captadas las señales se miden una serie de parámetros (ángulos, energías, tiempo...) y a partir de ahí se calcula la posición desconocida del elemento móvil a partir de las posiciones y/o zonas de los AP's y de los parámetros medidos relativos a esos nodos.

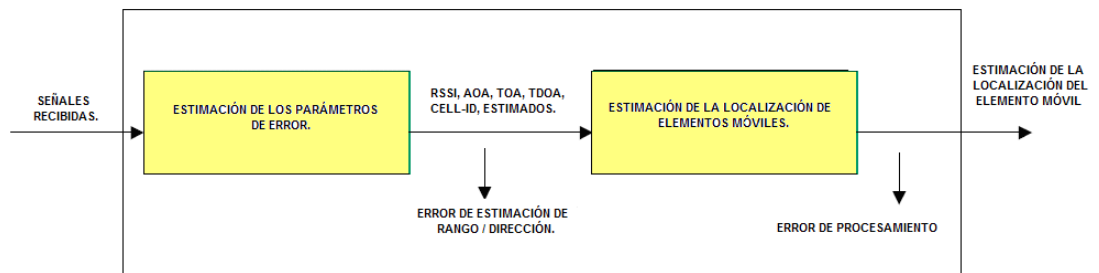


Figura 1: Diagrama de bloques de un Sistema de Localización.

2.1.2 Clasificación de los sistemas de localización.

Los sistemas de localización se pueden clasificar en función de diferentes parámetros [Ruiz, 2005]:

- Si la posición del móvil es conocida o no:
 - **Localización local:** Se conoce la posición inicial del móvil y se hace un seguimiento para determinar la siguiente posición.
 - **Localización global:** Se intenta conocer la posición del móvil sin tener información de la localización anterior.
- Según se representa la posición:
 - **Sistemas deterministas:** Predicen, con el error más bajo posible, la posición siguiente del dispositivo a partir de la posición actual del sistema, las observaciones que van captando los sensores y los cambios que se producen en el entorno de localización. Un ejemplo de la técnica es la correlación cruzada.
 - **Sistemas probabilísticos:** Son los sistemas en los que las variables que determinan la localización se ven afectadas por otras variables aleatorias como el ruido y la localización se determina en función de una probabilidad. Un ejemplo de técnicas son las redes neuronales.

- En función del tipo de entorno en el que actúe:
 - **Sistema de localización para exteriores (outdoor)**, por ejemplo GPS, Loran...
 - **Sistema de localización en interiores (indoor)** por ejemplo, localización GSM o Wi-Fi
- En función del tipo de parámetros de medida que utilice:
 - **TOA**: tiempos de vuelo
 - **AOA**: Medidas de ángulos
 - **RSSI**: Intensidad de la señal recibida.
 - **Cell-Id**: Medida de proximidad, donde se calcula si se encuentra dentro de una zona de cobertura.
- En función del algoritmo de cálculo de posición que tenga implementado el sistema de localización:
 - **Algoritmos de multiecuación para TOA**, como métodos algebraicos, con los cuales se obtiene la localización operando un sistema de ecuaciones.
 - **Triangulación** que es la técnica que maneja AOA.
 - **Detección por proximidad**, se identifica al dispositivo con la zona del AP y/o nodo más cercano.
- Según las coordenadas de localización que emplee:
 - **Localización absoluta**: Localización respecto a un sistema de referencia general.
 - **Localización relativa**: Localización relativa a varios AP's que forman parte del sistema sensorial.
- Granularidad:
 - **Localización física**: La localización consiste en un conjunto de coordenadas cartesianas en 2D o en 3D.
 - **Localizaciones simbólicas**: Indica si el dispositivo a localizar se encuentra en una zona determinada.
- Precisión:
 - **Precisión centimétrica**: Se consigue en técnicas ultrasónicas o en UWB.
 - **Precisión métrica**: GPS, Bluetooth, WLAN.
- Necesidad de desarrollo de Hardware a medida:
 - Sistemas de localización que requieren equipos específicos para la transducción y procesamiento de los datos.
 - Sistemas de localización que aprovechan la estructura existente y más un software adicional, completan dicha función.
- Movilidad de nodos:
 - AP's y/o nodos **estáticos**.
 - AP's y/o nodos **dinámicos**.
- Dónde se calcula la posición:
 - Método **centralizado**: Un sistema recibe la información de los diferentes AP's y/o nodos que han recibido la señal del dispositivo a localizar y procesa dichos datos para obtener las coordenadas o zona óptima donde se encuentre el elemento desconocido.

- Método **distribuido**: En este método es el propio dispositivo el que a partir de la información que recibe de los AP's y/o nodos calcula su propia localización. Éste método favorece a la privacidad ya que la posición en la que se encuentra, es conocida únicamente por el propio dispositivo.
- Tipo de energía utilizada:
 - **Señales ópticas**: IR, Láser.
 - **Acústicas**: Ultrasónicas.
 - **Señales radio**: WLAN, GPS, TV, Bluetooth.

2.2 TECNOLOGÍAS INALÁMBRICAS

En este apartado se van a explicar las diferentes tecnologías inalámbricas que se pueden utilizar en un sistema de localización para redes LAN (Local Area Network), Personal Area Network (PAN) y WAN (Wide Area Network). Entre las tecnologías que se van a explicar, no se encuentra GPS, porque aunque en exteriores, es la mejor tecnología de posicionamiento, en entornos indoor es bastante ineficiente y este proyecto se centra en sistemas de localización de este último tipo.

Una de las razones por las que GPS no es compatible con un escenario indoor, es que requiere línea directa de visión entre la estación base y el usuario a localizar. Debido a las múltiples reflexiones que puede sufrir la señal al chocar contra los posibles obstáculos que hay dentro de un edificio (paredes, muebles, personas...), es difícil conseguir un único camino entre los participantes de la comunicación.

Otra desventaja es que la señal procedente del satélite llega muy atenuada al receptor indoor y por lo tanto la información que se podría obtener no sería de buena calidad, al igual que la localización resultante del procesamiento de estos datos.

2.2.1 Infrarrojos (IR)

2.2.1.1 Definición

La radiación infrarroja [Fontal, 2006] es un tipo de radiación electromagnética de mayor longitud de onda, λ , que la luz visible pero menos que la de las microondas. Por este motivo, tiene menor frecuencia que la luz visible y mayor que las microondas.

2.2.1.2 Historia

El nombre de infrarrojo [ConocimientosWeb.com, 2005] que significa “por debajo del rojo”, se observó por primera vez al dividir la luz solar en diferentes colores por medio de un prisma que separaba la luz en su espectro, de manera que en ambos extremos aparecen visibles las componentes del rojo al violeta. Aunque estos experimentos habían sido realizados anteriormente por Isaac Newton, William Herschel observó en el año 1800 que se recibía radiación por debajo del rojo al situar medidores de calor en las diferentes zonas no visiblemente irradiadas por el espectro.

Su longitud de onda, entre 700 nanómetros y 1 milímetro, es la siguiente en longitud al rojo, el color de longitud de onda más larga de la luz visible. Su espectro va en términos generales desde 3×10^{11} hasta 2×10^{14} Mhz.

Son útiles para comunicaciones locales punto a punto de corto alcance (inferior a 10m) o para aplicaciones multipunto en una zona pequeña de cobertura como es el caso de la localización. Fue la primera tecnología empleada para este fin.

Los sistemas que utilizan IR para la localización de dispositivos móviles, utilizan dispositivos especiales denominados tags que emiten radiación infrarroja en modo difuso, es decir, de forma radial. Las antenas tienen sensores ópticos que detectan los “tags” que están cerca e informan al servidor de localización. Se puede considerar más un sistema de detección que de localización ya que la posición del elemento etiquetado con el tag IR, se obtiene de la posición fija y conocida de los sensores que detectan los tags.

2.2.1.3 IrDA (Infrared Data Association)

IrDA [Caviedes, 1999] es la organización que establece normas internacionales para el hardware y el software empleados en comunicaciones infrarrojas. Es muy importante en comunicaciones inalámbricas. Fue creada en 1993 por HP, IBM, Sharp, entre otros. También se le ha denominado así a la tecnología creada por la organización.

Los estándares IrDA definen comunicaciones punto a punto sobre un haz de luz infrarroja que requiere línea directa de visión, un ángulo inferior o igual a 30° y un alcance de 1 m. para obtener velocidad de transmisión de datos entre 9.6 Kbps y 16 Mbps en función del entorno.

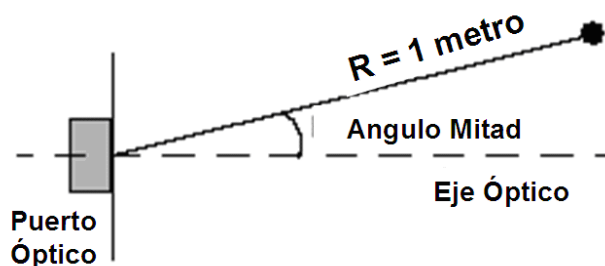


Figura 2: Comunicación punto a punto del estándar IrDA.

2.2.2 Ultrasonido

2.2.2.1 Definición

Los ultrasonidos [Téllez] son ondas sonoras, de frecuencia superior a la que el oído humano puede percibir normalmente (el rango de frecuencia que podemos percibir oscila de 16 Hz a 20 KHz).

También se podría definir como un tren de ondas mecánicas, generalmente longitudinales (onda sonora), originadas por la vibración de un cuerpo elástico, y propagadas por un medio material.

2.2.2.2 Historia

Galtan, en el año 1883, estudió el rango de frecuencias que el oído humano es capaz de oír [Téllez]. Gracias a este análisis, puedo concluir que la propagación de las ondas no audible por el ser humano, era similar al resto de ondas sonoras, exceptuando que eran más absorbidas por el aire.

Desde este momento, se han seguido realizando numerosas investigaciones relacionadas con la generación de ultrasonidos. A continuación, se indican algunos de los nombres más destacados.

- En la década de los 80 del siglo XIX, los hermanos Curier descubrieron la piezoelectricidad y Lippmar y Voigt fueron quienes trabajaron con el efecto piezoeléctrico inverso, aplicable a la generación de pulsos.
- Joule en 1847 y Pierce en 1928 descubrieron el efecto magnetoestrictivo, directo e inverso.
- Durante el siglo XX, Langervin los empleó para sondeos subacuáticos en la I Guerra Mundial, Richard y Fessenden, en la década de los 1910 localizaron icebergs mediante el método de impulsos y Mulhouser y Fierestone entre 1933 y 1942 aplicaron los ultrasonidos a la industria e inspección de materiales.

2.2.2.3 Funcionamiento

Los ultrasonidos tiene múltiples aplicaciones que se pueden diferenciar según su potencia de emisión.

- Alta potencia (mW a Kw): Sonar, limpieza de canalizaciones, soldadura, erosión, cirugía, etc.

- Baja potencia: Ecografía, localización de defectos en objetos, superconductividad, robótica para medidas de distancia, captación y mapeado de entornos, detección de obstáculos.

Para cada uno de ellos, la forma en que se hace uso de dichas ondas sonoras es diferente. En el caso de la localización, se centra en el funcionamiento básico de los ultrasonidos como medidores de distancia.

En este caso, se tiene un transmisor que emite un pulso de ultrasonido que rebota sobre un determinado objeto. La reflexión de ese pulso es detectada por un receptor de este tipo de ondas sonoras. La mayor parte de sensores de ultrasonido de bajo coste se basan en la emisión de un pulso de ultrasonido, cuyo lóbulo es la forma cónica. Una vez que se obtiene el tiempo (t) transcurrido entre la emisión de la onda sonora y la recepción del eco se puede calcular la distancia a la que se encuentra el obstáculo con la siguiente fórmula donde V es la velocidad del sonido en el aire ~ 340 m/s:

$$d = \frac{1}{2} \times V \times t$$

2.2.3 RFID

2.2.3.1 Definición

RFID (Radio Frequency Identification) [Systems., 2005], o lo que es igual Identificación por Radiofrecuencia (RF), es un método de identificación automático que almacena y recupera datos de forma remota sin necesidad de contacto, ni siquiera visual a través de una etiqueta RFID (tag) o un transpondedor.

Los sistemas RFID tienen tres elementos principales:

1. Un lector de tags: dispositivo que tiene una o más antenas de instalación fija o móvil que emiten y/o reciben ondas de radio que activan una etiqueta en el campo de lectura, provocando que ésta refleje su información en el lector.
2. Una base de datos: ordenador donde se almacena y trabajan los datos leídos.
3. Un transpondedor: etiqueta electrónica que se compone de un microchip y de una antena. El microchip ó memoria de datos puede ser leído y escrito sin contacto, a distancia sin línea de vista directa vía ondas de radio, usando la antena que forma el tag RFID. Los microchips que son regrabables, permiten modificar la información que contienen, por ello es útil para realizar seguimientos de objetos que portan etiquetas.

2.2.3.2 Historia

El origen de RFID data de la II Guerra Mundial [TOSHIBA], cuando el radar que usaban los alemanes no les permitía diferenciar los aviones enemigos de los suyos propios. Sin embargo, los alemanes descubrieron que si los pilotos balanceaban sus aviones al volver a la base, la señal reflejada cambiaría y conseguirían discernir entre los aviones de los dos ejércitos. Éste fue el primer RFID pasivo.

Durante los años 50 y 60 se siguió investigando en los sistemas radar y de comunicaciones por radiofrecuencia, para estudiar como identificar objetos a distancia.

Las primeras patentes para dispositivos RFID fueron solicitadas en EE.UU., en enero de 1973, cuando Mario W. Bandullo presentó una tarjeta RFID activa con memoria reescribible. El mismo año, Chales Walton recibió la patente de un sistema RFID pasivo que abría las puertas sin llaves. Durante los años 70, EE.UU. también trabajó para el desarrollo de esta tecnología. Posteriormente se han ido mejorando la calidad de transmisión, recepción y distancia.

2.2.3.3 Funcionamiento

El lector envía ondas de radiofrecuencia o señal de interrogación a los tags presentes (TOSHIBA) (Urraza & YAEL, 2007) que éstos captan a través de su antena. Estas ondas activan al microchip que

mediante la micro antena y la radiofrecuencia devuelven (sistema pasivo) o emiten (sistema activo) una señal al lector indicando cuáles son sus respectivos identificadores únicos.

Una vez que el lector obtiene los códigos únicos, los transmite a una base de datos, donde previamente se han almacenado las características de los elementos que portan el tag RFID, que permite consultar la identidad de los mismos.

A continuación se explica el proceso de forma más técnica:

1. El lector genera un campo de radiofrecuencia, normalmente conmutando una bobina a alta frecuencia. Las frecuencias usuales van desde 125 Khz. hasta la banda ISM de 2.4 Ghz o superior.

2. El campo de radiofrecuencia genera una corriente eléctrica sobre la bobina del receptor. Esta señal es rectificada y de esta manera se alimenta el circuito. Usa la misma frecuencia como reloj.

3. Cuando la alimentación sea suficiente el circuito transmite sus datos.

4. Un protocolo anticolidión permite gestionar la respuesta simultánea de múltiples RFID. En el caso que se dé colisión, el lector lo detecta, para la transmisión de las tarjetas durante un tiempo y posteriormente cada tarjeta por separado responde por medio de un algoritmo de espera.

5. El lector detecta los datos transmitidos por la tarjeta, como una perturbación del propio nivel de la señal. La señal recibida por el interrogador desde la tarjeta está a un nivel de -6° dB por debajo de la portadora de transmisión. El rango de lectura para la mayoría de los casos está entre los 30 y 60 cm. de distancia entre lector y tarjeta.

6. La información recibida se integra con el resto del Sistema de Información.

Normalmente el sistema de modulación es AM con codificación Manchester NRZ. Se aplican algunos sistemas como división de la frecuencia del reloj de recepción para conseguir mayor alcance y más inmunidad al ruido eléctrico. La mayor parte de los sistemas tienen una memoria EEPROM donde se almacenan datos. También se puede utilizar cifrado de clave pública para conseguir mayor seguridad ante posibles escuchas inadecuadas que no pertenezcan al sistema.

2.2.4 Ultra Wide Band (UWB)

2.2.4.1 Definición

Ultra Wide Band, conocido también como UWB o como Digital Pulse Wireless [Barrett, 2000] [Wikipedia, 2007], es una tecnología inalámbrica para transmisiones de grandes cantidades de datos sobre un amplio espectro de bandas de frecuencia con baja potencia para distancias cortas. También tiene la capacidad de llevar señales a través de puertas y otros obstáculos que tienden a reflejar señales en anchos de banda más limitados y con mayor potencia.

2.2.4.2 Historia

El primer pulso basado en radio UWB fue Spark Gap, el primer desarrollo por Guglielmo Marconi a finales de 1800 para transmitir código Morse. La tecnología Spark Gap fue dominante hasta la aparición de radios de banda estrecha (ondas continuas) que fueron propuestas como la mejor opción para comunicaciones de voz en los primeros años de 1900. Cerca de 1924, Spark Gap radios fue prohibido en muchas aplicaciones debido a la irregularidad de sus emisiones RF. Fue sustituido por radio base en portadoras (ancho de banda estrecha).

En 1962, Hewlett-Packard desarrolló el osciloscopio, que permitió observar y medir la respuesta al impulso (pulse based) de redes de microondas. En los 60, los avances del estudio sobre esta tecnología, descubrieron una nueva forma de transmisión para radar e intercambio de información mediante pulsos.

En 1973, se concedió el primer ancho de banda patentado a Sperry Research Center. A partir de ese momento, las tecnologías que funcionaban sobre un gran ancho de banda vuelven a emerger.

A finales de 1980, la tecnología UWB se definió como tecnología banda base de libre portadora o impulso, pero no fue hasta 1989 cuando el Departamento de Defensa de EE.UU., definió el término “Ultra WideBand.”

Durante 30 años se han desarrollado técnicas usando esta tecnología, hasta que recientemente las aplicaciones UWB se permitieron solo bajo licencias especiales. En Abril de 2002 FCC, la agencia federal del Gobierno de EE.UU. encargada de regular y administrar las materias de telecomunicaciones, publicó las Normativas UWB, permitiendo emisiones UWB sujetas a frecuencias y potencias limitadas, lo que atenúa el riesgo de interferencias a aquellos que comparten el mismo espectro. UWB intenta que se regularice mundialmente. El único país donde son legales los productos de consumo UWB es EE.UU. Intel está trabajando de cerca con los gobiernos locales para promover y facilitar las normativas de UWB adoptadas por FCC.

2.2.4.3 Funcionamiento

UWB se aceptó como un sistema basado en pulsos [Wikipedia, 2007], donde cada antena emite (antena transmisora) un pulso que ocupa al menos 500 Mhz o el 25% de la frecuencia central [Sinan Gezici, 2005]. La frecuencia de repetición de los pulsos puede ser muy baja (1 a 100 Mega pulsos/segundo) o muy alta (1 a 2 Giga pulsos/segundo).

El correspondiente receptor transforma los pulsos en datos para buscar una secuencia de pulsos conocida enviada por el transmisor.

Los pulsos que se envía en este sistema ocupan todo el ancho de banda, como consecuencia son muy cortos en espacio (menos de 60 cm. para un pulso de anchura de 500 Mhz y menos de 23 cm. para un pulso cuyo ancho de banda sea 1.3 Ghz) y las reflexiones de la señal no solapan al pulso original lo que provoca que disminuya el efecto del multipath aunque no el ISI (Interferencia entre Símbolos), que se podría mitigar con técnicas de codificación.

Una diferencia entre las transmisiones radio y transmisiones radio UWB es que en los sistemas tradicionales la información se envía variando la fase, amplitud o frecuencia de una onda sinusoidal. Sin embargo en transmisiones UWB se puede transmitir información generando energía radio en unos instantes específicos de tiempo y ocupando un gran ancho de banda permitiendo un pulso-posición o tiempo-modulación.

La información también puede ser modulada en señales UWB (pulsos), codificando la polaridad de los pulsos, la amplitud y/o usando pulsos ortogonales.

2.2.5 Bluetooth

2.2.5.1 Definición

Bluetooth [Gruop, 2000] es una tecnología inalámbrica de onda de radio de corto alcance que opera en la banda de frecuencia ISM (Industrial, Científica y Médica), que no requiere licencia y que se encuentra concretamente entre 2.4 Ghz y 2.485 Ghz. Bluetooth es el nombre común de la especificación de IEEE 802.15.1, que define un estándar global de transmisión de información (voz y datos) inalámbrico entre diferentes dispositivos, a través de un enlace de radiofrecuencia seguro y sin licencia de corto rango.

2.2.5.2 Historia

El nombre Bluetooth [Wikipedia, Bluetooth, 1994] deriva del rey danés y noruego Harold Blontand cuya traducción al inglés es Harold Bluetooth, conocido por unificar las tribus noruegas, suecas y danesas, ya que el objetivo de las empresas que comenzaron a estudiarla también era unificar, pero en este caso, las diferentes tecnologías de los ordenadores, teléfonos móviles y el resto de periféricos. De hecho el símbolo del Bluetooth es la unión de las runas nórdicas H y B.

En 1994, Ericsson inició un estudio de una nueva interfaz de bajo coste y consumo que permitiera interconectar teléfonos móviles y otros accesorios, sin la necesidad de utilizar hilos (vía radio), que aunque el uso de cables es eficiente, su instalación es engorrosa y limita la movilidad de los usuarios.

Este estudio era la continuación de un proyecto que investigaba multicomunicadores conectados a una red celular, hasta que se consiguió un enlace de radio de corto alcance, llamado MC link. Actualmente se basa en un chip de radio, lo que permitirá su utilización en un gran número de aplicaciones.

En 1999, se fundó SIG (Special Interest Group) de Bluetooth. Es un grupo de empresas que trabajan conjuntamente para desarrollar, promover, definir y publicar las especificaciones de esta tecnología inalámbrica a corta distancia. En febrero de 1999, los componentes que fundaron el SIG eran:

- Ericsson Mobile Communications.
- Intel Corporation
- IBM Corporation
- Toshiba Corporation
- Nokia Mobile Phones

En julio de 1999 se publicó la versión 1.0 de las especificaciones de Bluetooth. En diciembre del mismo año, el núcleo del SIG tuvo cuatro grandes empresas más:

- Microsoft
- Lucent
- 3COM
- Motorola

En Abril del 2000 el SIG estaba formado por más de 1800 empresas.

Las compañías que forman el SIG pueden dotar a sus productos de la tecnología Bluetooth, aparte de conocer las especificaciones de la tecnología y usar libremente la banda de 2.4 Ghz, al contrario que las empresas externas que no pueden utilizar la tecnología al no tener la patente.

2.2.5.3 Funcionamiento

El estándar Bluetooth utiliza la técnica FHSS que consiste en dividir la banda de frecuencias de 2.402 a – 2.480 GHz en 79 canales de 1Mbps. Y posteriormente transmite la señal utilizando una secuencia de canales conocidos por ambos extremos de la comunicación. Al cambiar de canal 1600 veces por segundo, permite evitar las interferencias de las otras señales que trabajan en su mismo rango.

La red Bluetooth que se forma al conectarse dos dispositivos Bluetooth y aquellos que se encuentren dentro de un rango, se denomina piconet. El modo de operación en el que se basa es maestro/esclavo.

El proceso que tienen que seguir dos usuarios que quieran iniciar una comunicación Bluetooth es el siguiente:

1. Durante el tiempo en el que el usuario no quiere iniciar ninguna comunicación Bluetooth, el dispositivo se encuentra en modo pasivo, donde sólo escucha la red.
2. El establecimiento de una conexión bluetooth, comienza con la fase de solicitud, en el que el maestro envía una petición a todos los usuarios que se encuentran en su mismo rango, denominados puntos de acceso. Los dispositivos que reciben la solicitud, responden con su dirección. A continuación, el maestro sincroniza su reloj y frecuencia con el punto de acceso que ha elegido. Esta técnica se llama paginación.
3. Así, se establece un enlace entre el maestro y el punto de acceso. A continuación, se inicia el proceso de descubrimiento del servicio a través del protocolo SDP, que determina los servicios Bluetooth disponibles y sus características en el dispositivo esclavo.
4. Una vez finalizada esta fase, el maestro crea un canal de comunicación con el punto de acceso esclavo, mediante el protocolo L2CAP.
5. Algunos puntos de acceso pueden incluir el mecanismo de seguridad conocido como emparejamiento. El dispositivo esclavo envía al maestro una petición de emparejamiento y éste

último tiene que escribir la clave, denominada PIN, del punto de acceso trasmisor. Si la clave es correcta, se realiza la conexión.

2.2.6 WLAN

2.2.6.1 Definición

Es un sistema de comunicación de información que transmite y recibe datos utilizando ondas electromagnéticas entre dispositivos, como sustitución de las redes de par trenzado, coaxial o fibra óptica que se usa en las LAN [Patmathan, 2006:008] [Wikipedia]. WLAN permite conectividad inalámbrica peer to peer, dentro de un edificio o de pequeñas zonas al aire libre.

2.2.6.2 Historia

Los primeros estudios y experimentos que publicó IEEE sobre comunicaciones inalámbricas [Tutorial-Report.com, 2005:07] datan en 1979, en manos de ingenieros de IBM en Suiza que pretendían crear una LAN con enlaces infrarrojos. Las investigaciones sobre comunicaciones sin hilos continuaron utilizando tanto la tecnología infrarroja como microondas. En 1985, FCC asignó las bandas ISM, 902-928MHz, 400-2, 48355GHz, 5, 725-5, 850 GHz reservado para las redes inalámbricas, basadas en Espectro ensanchado (SS), con las opciones Secuencia directa (DS), o Salto de frecuencia (FH).

En mayo de 1991, se consiguió desarrollar experimentos WLAN, que superaban 1Mbit/s de velocidad que es el umbral establecido por IEEE para que la red se considere LAN, como aplicación empresarial.

En 1989 en el seno de IEEE 802, se forma el comité 802.11 para estandarizar la normativa de WLAN, que no se finaliza hasta 1999, aunque fue en 1997 cuando IEEE detalló las especificaciones de 802.11.

A partir de 1992, se crea un grupo que potencia las comunicaciones inalámbricas, ya sea mediante infrarrojos ó microondas, como por ejemplo IrDA (Infrared Data Association), WLI Form, Wireless LAN Interoperability Forum, etc.

Actualmente, existen cuatro estándares del grupo 802.11 que son: 802.11 original, 802.11a, 802.11b, 802.11g, 802.11i, 802.11e. Como todos los estándares 802 de IEEE, el estándar 802.11 se centra en los dos niveles inferiores de OSI, la capa física y la capa de enlace, así que cualquier aplicación LAN, sistema operativo de red, y/o protocolo (TCP/IP), se podrá ejecutar en WLAN tan fácilmente como sobre Ethernet.

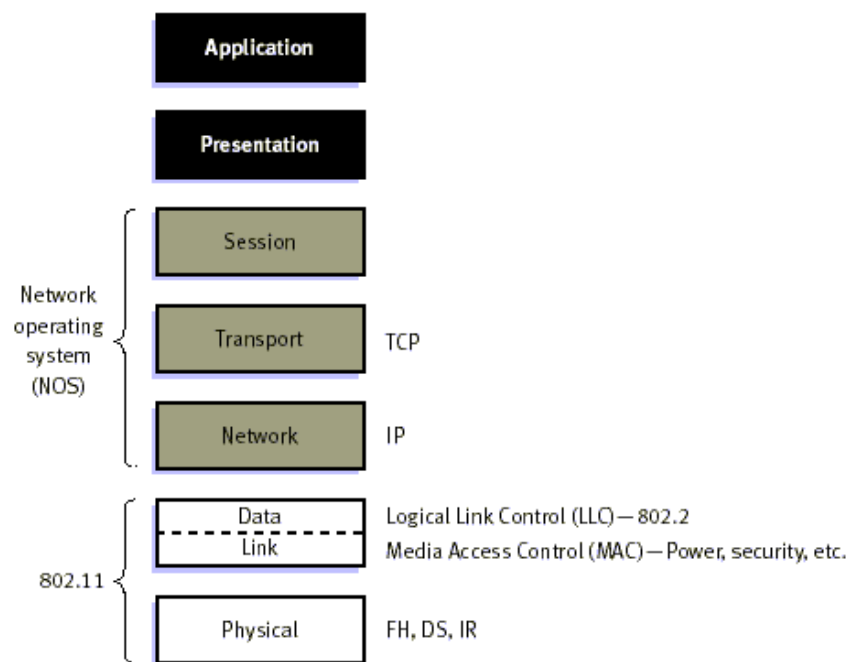


Figura 3: Niveles del estándar 802.11

2.2.6.3 Arquitectura 802.11

Cuando dos o más estaciones móviles se quieren comunicar entre sí forman lo que se conoce como una BSS (Basic Service Set) [Wikipedia].

Una BSS que no se conecta a otra BSS, se denomina IBSS (Independent Basic Service Set) o se refiere como una red ad hoc. Una red ad hoc es una red donde las estaciones sólo se comunican peer to peer, no hay nadie que dé permisos para “hablar” y la mayor parte de estas redes son espontáneas y pueden establecerse rápidamente.

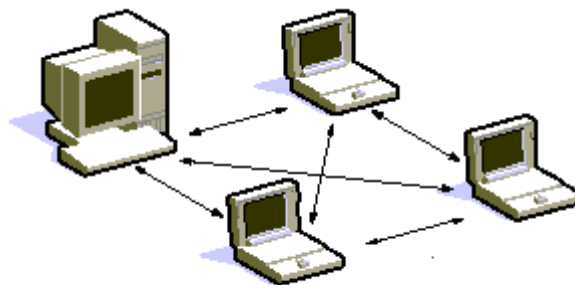


Figura 4: Independent Basic Service Set (IBSS)

Cuando las BSS's se interconectan, la red se convierte en una red con infraestructura o en modo infraestructura. Una infraestructura 802.11 tiene varios elementos. Dos o más BSS's se interconectan usando un Sistema de Distribución o DS. Este concepto incrementa la cobertura de la red. Cada BSS se convierte en un componente de una extensa y gran red. El acceso a un DS se consigue a través de un punto de acceso (AP). Un AP es una estación que tiene capacidad de enrutamiento, es decir, los datos se mueven entre los BSS's y los DS's con ayuda de los AP's.

Al crear una red grande y compleja, formada por BSS's y DS's se puede hablar del siguiente nivel jerárquico que es ESS (Extended Service Set). Una característica del ESS es que la red entera parece

como una IBSS para la capa LLC (Logical Link Control Layer). Esto significa que las estaciones dentro de los ESS se pueden comunicar o incluso mover entre BSS's de forma transparente para el LLC.

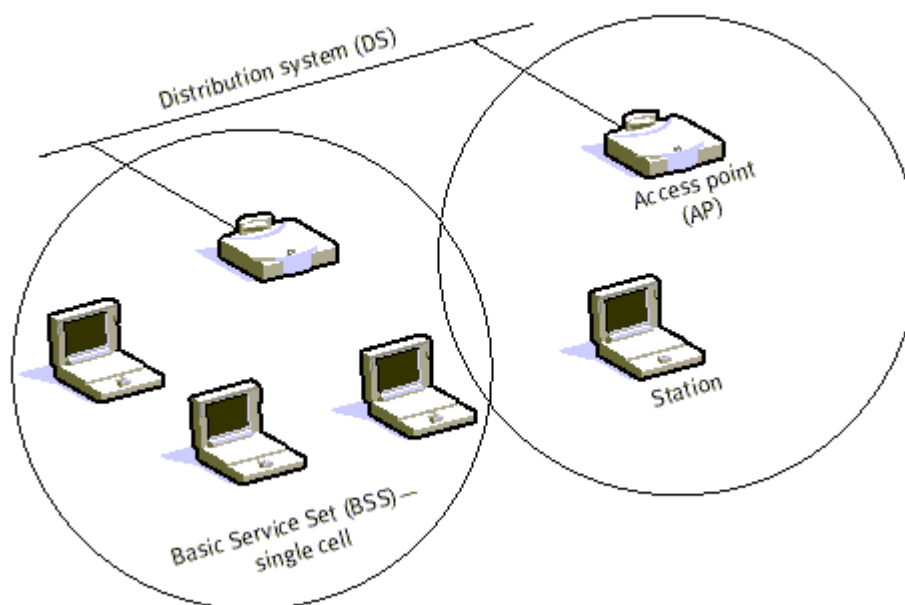


Figura 5: Extended Service Set (ESS) – multiple cells

Uno de los requerimientos de IEEE 802.11 es que puedan interactuar con redes cableadas. Para ello, 802.11 crea la figura denominada portal que es una integración lógica entre las LAN's inalámbricas y por cable. También puede actuar como punto de acceso de un DS. Todos los datos que se intercambien entre la red 802.11 inalámbrica y 802.11 cableada tienen que pasar por un portal.

Capa Física

Las tres capas físicas definidas por 802.11 [Tutorial-Report.com, 2005:07] incluyen dos técnicas radio de espectro expandido y una de difusión de infrarrojos.

Los estándares basados en radio operan dentro de la banda ISM de 2.4 Ghz. por lo que los productos basados en 802.11 no requieren usar licencias o entrenamiento especial.

Las técnicas de espectro expandido, acorde con los requerimientos de regularización, incrementan la confiabilidad, aumenta el tiempo de procesamiento y permite que muchos productos sin relación compartan el espectro con mínimas interferencias.

El estándar inalámbrico original 802.11 define un ratio (capacidad) de datos de 1 Mbps y 2 Mbps usando salto de frecuencia de espectro expandido (FHSS) o secuencia directa de espectro expandido (DSSS). Es importante destacar que FHSS y DSSS son diferentes en el mecanismo de la señalización.

La técnica de salto de frecuencia, divide la banda de 2.4 GHz en 75 subcanales de 1 Mhz. Las estaciones origen y destino se rigen según un patrón de salto, y los datos se envían sobre una secuencia de subcanales. Cada conversación dentro de la red 802.11 ocurre sobre un patrón de salto diferente. Los patrones se diseñan para minimizar la ocasión que dos estaciones usen el mismo canal simultáneamente. La técnica FHSS se puede usar para un modelo de radio relativamente sencillo, porque la velocidad máxima que alcanza es de 2 Mbps. La limitación viene establecida por FCC que restringe el ancho de banda de los subcanales a 1 Mhz.

La técnica de señalización de secuencia directa divide la banda de 2.4 GHz en 14 canales de 22 Mhz. Estos canales se solapan unos a otros parcialmente, excepto tres de ellos que no se solapan. Los datos se envían a través de uno de los canales de 22 MHz sin saltar a otro canal. Para compensar el ruido en un

canal radio, se usa una técnica llamada “chipping”. Cada bit de los datos de usuario se convierte en una serie de patrones de bit redundantes que se llaman chips. La redundancia de cada chip combina con la separación de la señal que va en el canal de 22 MHz que proporciona chequeo y corrección de errores. Incluso si parte de la señal se degrada, en muchos casos, se puede recuperar, minimizando la necesidad de retransmisiones.

Capa de enlace de datos

La capa de enlace de datos dentro de 802.11 [Tutorial-Report.com, 2005:07] consiste en dos subcapas: LLC (Logical Link Control) y MAC (Medium Access Control). 802.11 usa el mismo LLC que 802.2 y 48 bits de direccionamiento como cualquier otra LAN 802. Esto permite el traspaso de redes inalámbricas a redes IEEE cableadas. La MAC es única para WLAN's. La MAC de 802.11 es muy similar en concepto a 802.3. Se diseña para soportar múltiples usuarios en un medio compartido. Para 802.3 Ethernet LAN's, CSMA/CD regula como las estaciones de Ethernet establecen acceso al cable y como éstos detectan y manejan la colisión que ocurre cuando dos o más dispositivos intentan simultáneamente comunicar sobre la LAN. En un 802.11 WLAN la detección de colisiones no es posible debido al problema que se conoce como “near/far”, es decir, en sistemas radio la estación pierde la capacidad de escuchar la colisión.

802.11 usa un protocolo conocido como CSMA/CA o DFC. CSMA/CA permite evitar colisiones usando paquetes explícitos de confirmación (ACK), que los envía la estación receptora para confirmar que el paquete de datos llegó sin errores.

CSMA/CA trabaja de la siguiente forma: Una estación desea transmitir, si no detecta actividad, la estación espera un período de tiempo aleatorio y después transmite si el medio aún está libre. Si el paquete se recibe intacto, la estación receptora edita una trama ACK que, una vez que la estación origen lo recibe con éxito, se completa el proceso. Si la estación origen no detecta el ACK, eso significa que el paquete de datos que envió no llegó con éxito al destino o el ACK se recibió corrupto, es decir, se asume que ha habido una colisión y la información se transmite de nuevo, después de esperar un período de tiempo aleatorio.

CSMA/CA proporciona un camino de compartición del medio. Este mecanismo también maneja interferencias y otros problemas de radio. Sin embargo, agrega a 802.11 algunas cabeceras que 802.3 no tiene. Por este motivo 802.11 tiene un funcionamiento más lento que Ethernet LAN.

Otro problema de la capa MAC específico para WLAN es el efecto “hidden node”, en el cual dos estaciones en lados opuestos de un AP pueden ambos escuchar actividades de un AP, pero no actividades de la otra estación. Esto hecho puede ser debido a la distancia o una obstrucción. Para solventar este problema, 802.11 especifica una opción, RTS/CTS (Request To Send/Clear To Send), al protocolo de la capa MAC. Cuando este proceso está funcionando, una estación origen transmite un RTS y espera para que el AP responda con un CTS. Todas las estaciones de la red pueden escuchar el CTS del AP, que les causa un retardo en cualquier intento de transmisión. Esto permite que la estación origen pueda transmitir y recibir la confirmación sin peligro a la colisión. RTS/CTS añade cabeceras adicionales a la red, reservando el medio temporalmente, se usa normalmente en paquetes de gran tamaño. Esto daría lugar a utilizar mayor ancho de banda.

Finalmente, la capa MAC 802.11 proporciona dos características más que ofrecen robustez: CRC Checksum y fragmentación de paquetes. Cada paquete tiene un CRC Checksum calculado para asegurar que los datos no se dañen durante el camino de la transmisión. Éste es diferente que en Ethernet, ya que los protocolos de capas superiores como TCP tienen la funcionalidad de chequear errores.

La fragmentación de paquetes permite que los paquetes grandes se dividan en unidades más pequeñas cuando se envían por el aire, ya que los paquetes de gran volumen de datos se dañan con mayor facilidad en entornos muy congestionados o en medios con un alto factor de interferencias. Esta técnica reduce la necesidad de retransmisiones en muchos casos y esto aumenta el rendimiento de la red inalámbrica.

La capa MAC es responsable de reensamblar los paquetes recibidos. Es un proceso transparente para capas de niveles superiores.

2.2.6.4 Estándar IEEE 802.11

El ratio de datos que soporta el estándar original de 802.11 [Tutorial-Report.com, 2005:07] es demasiado lento para soportar los requerimientos de negocio y como consecuencia limita la demanda de WLAN. Para satisfacer la necesidad de soportar un ratio de datos más elevado, IEEE creó el estándar 802.11b (también conocido como 802.11 High Rate) que alcanza hasta 11Mbps. Después de 802.11b, surgió 802.11a, que soporta hasta 54 Mbps, y en enero del 2002 se aprobó 802.11g.

IEEE 802.11b

La arquitectura básica, funcionalmente y servicios de 802.11b se definen por el estándar original 802.11. La especificación 802.11b afecta sólo a la capa física, añadiendo mayor ratio de datos y mayor robustez en la conectividad.

La clave de la contribución de 802.11b al estándar WLAN consistió en estandarizar la capa física soportando dos nuevas velocidades: 5.5 Mbps y 11 Mbps. Para lograr esto, DSSS se seleccionó como la única técnica de la capa física, violando las normativas actuales de FCC. La implicación es que el sistema 802.11b funcionará a 1 Mbps y 2 Mbps con sistemas DSSS, pero no con FHSS.

El estándar DSSS 802.11 original especifica un chipping de 11bits que se denomina secuencia Barrer que codifica todos los datos que se envían sobre el aire. Cada secuencia chip representa un bit de datos simples (1 ó 0) y se convierte a una forma de onda, llamado símbolo que se puede enviar sobre el aire. Estos símbolos se transmiten a 1 MSímbolo (1 millón de símbolos por segundo), para ello se usa la técnica de modulación BPSK. En el caso de 2Mbps, la implementación es QPSK. Doblar el ratio de datos disponible en BPSK, es una vía para aumentar la eficiencia del ancho de banda. Para incrementar el ratio de datos en el estándar 802.11b, se emplean técnicas avanzadas de codificación.

802.11b especifica Complementary Code Keying (CCK), que consiste en un set de 64-8 bit palabras código. Estas palabras código tienen propiedades matemáticas únicas, que les permite ser correctamente diferenciadas unas de otras por un receptor, incluso con la presencia de una gran cantidad de ruido e interferencias multipath.

El ratio de 5.5 Mbps usa CCK para codificar 4 bits por portadora, mientras que el ratio de 11 Mbps codifica 8 bits por portadora. Ambas velocidades usan QPSK como técnica de modulación y señal a 1.375 MSímbolos por segundo. Así se obtiene el mayor rango de ratio de datos.

802.11b usa también un ratio de datos dinámico, ajustados automáticamente para compensar la naturaleza cambiante del canal radio. Idealmente, los usuarios se conectan a una capacidad de 11Mbps, pero se puede ver modificada por interferencias que provienen de otras tecnologías de red como, microondas, teléfonos sin hilos 2.4 Ghz, y Bluetooth. Otra deficiencia de 802.11b es la falta de interoperabilidad con dispositivos de voz y de QoS para contenidos multimedia.

A parte de las limitaciones que se acaban de registrar, 802.11b es el más claro leader en negocio e institucional de redes inalámbricas y en clientes particulares.

IEEE 802.11a

802.11a es más rápido que 802.11b, con un máximo de ratio de datos a 54 Mbps que opera en la frecuencia de 5 Ghz y permite ocho canales simultáneamente. IEEE 802.11a tiene siete ratios más bajos que son: 48, 36, 24, 18, 12, 9 y 6 Mbit/seg.

802.11a usa OFDM, un nuevo esquema de codificación que ofrece beneficios sobre el espectro ensanchado en canales con disponibilidad. Los canales independientes son los disponibles y se convierten en la red inalámbrica más estable. OFDM define un total de 8 canales no solapados de 20 Mhz a través de dos paso bajo. A comparación con 802.11b, éste usa tres canales no solapados.

Las ventajas de 802.11a sobre 802.11b son, aparte del número de rangos de bajo ratio, el uso de una banda de frecuencias más alta, 5Ghz, ya que es una banda wider y está menos concurrida que la banda de 2.4 Ghz de 802.11b que comparte con microondas y dispositivos Bluetooth. Banda wider significa que

pueden coexistir más canales de radio sin interferencias. Cada canal de radio corresponde a una red separada o un segmento conmutado en la misma red.

Una de las grandes desventajas es que no es directamente compatible con 802.11b y requiere nuevos productos puente que puede soportar ambos tipos de redes. Otro gran inconveniente es que 802.11a sólo está disponible para un máximo de cuatro canales en Japón y no se aprovecha su uso en Europa, donde Hiperlan2 es el que se utiliza.

IEEE 802.11g

Como se ha comentado en el último punto, uno de los grandes inconvenientes es la falta de compatibilidad de 802.11a con la banda de 802.11b. Para conseguir la interoperabilidad entre ambas, IEEE desarrolló 802.11g, que debía extender la velocidad y rango de 802.11b.

El estándar opera completamente en la frecuencia de 2.4 Ghz., pero usa un mínimo de dos modos (ambos obligatorios) con dos modelos opcionales. El modo de acceso es el mismo modo CCK de 802.11b (de ahí la compatibilidad) y el modo OFDM usado por 802.11a, pero en este caso en la banda de frecuencia de 2.4 Ghz. El modo obligatorio CCK soporta 11Mbps y el modo OFDM hasta un máximo de 54 Mbps. Existen también dos modos que usa diferentes métodos para lograr un ratio de datos de 22Mbps que son: PBCC-22 (Packet Binary Convolutional Coding) de 6 a 54 Mbps y CCK-OFDM con un ratio máximo de 33 Mbps.

La ventaja más evidente de 802.11g es que mantiene compatibilidad con 802.11b y ofrece un ratio de datos más rápido comparable con 802.11a. Sin embargo el número de canales disponibles, no se incrementa, ya que está en función del ancho de banda. Otra desventaja es que al trabajar en la banda de 2.4Ghz, se ve afectada por un gran número de interferencias y como consecuencia nunca será tan rápido como 802.11a.

IEEE 802.11i

Caracterizado por una elevada seguridad de transmisión.

IEEE 802.11e

Especialmente diseñada para comunicaciones de voz.

2.2.6.6 Otras tecnologías que compiten con IEEE 802.11

Hiperlan 2

Hiperlan 2 [Tutorial-Report.com, 2005:07] es una tecnología de red inalámbrica que opera en la licencia libre de 5Ghz (5.4 a 5.7 Ghz) U-NII banda. Hiperlan se diseñó para llevar celdas ATM, paquetes IP, paquetes firewire y datos celulares de teléfonos móviles. Al igual que 802.11a es una forma de Ethernet inalámbrico, Hiperlan 2 es comúnmente reconocido como ATM inalámbrico.

Ofrece una velocidad de transmisión de 54 Mbps, usa el método de modulación OFDM (20 canales de 20 Mhz cada uno). La diferencia con 802.11a radica en la capa MAC, ya que Hiperlan 2, usa TDMA/TDD ofreciendo un ratio de 42 Mbits superior a los 31 Mbps que ofrece 802.11a que utiliza CSMA/CD.

Home RF

Home RF [Tutorial-Report.com, 2005:07] fue la primera tecnología de red inalámbrica en hogares. Apareció a mediados del 2000. Usa frecuencias de radio para transmitir datos sobre rangos de 75 a 125 pies.

HomeRF usa SWAP (Shared Wireless Access Protocol), el cual es un estándar híbrido, desarrollado por IEEE 802.11. SWAP puede conectar a 127 dispositivos de red y transmitir a velocidades superiores de 2 Mbps.

La mayor desventaja de una red HomeRF es la velocidad de datos de transmisión. Dos Mbps es suficiente para compartir e imprimir ficheros normales, pero es insuficiente para flujos media e impresión o transferencia de grandes ficheros gráficos.

HomeRF no interfiere con Bluetooth y es mejor para transmitir señales de voz.

2.2.6.7 Comparativa entre estándares 802.11x y tecnologías competidoras

A continuación se puede observar un cuadro comparativo entre los diferentes estándares de 802.11x y las tecnologías que compiten con ellos.

Aplicación	Key Tech	Velocidad de datos (Max/Media)	Ventajas	Desventajas
Enterprise Networking	802.11	2 Mbps/ 1.2 Mbps	Red inalámbrica de área local.	Lenta, cara e insegura
	802.11b	11 Mbps/5.5 Mbps	Rápida, barata y más fuerte que 802.11	La seguridad todavía no es muy fuerte y es más caro que el cable.
	802.11g	22 Mbps	Más rápida que 802.11b	Las tecnologías competidoras podrían dividir a los proveedores centrales.
Enterprise and Metropolitan Area Networking	802.11a	54 Mbps/24 Mbps, futuras iteraciones están pensadas para alcanzar 100 Mbps	Más rápida que 802.11b y 802.11g	Nuevo esquema de modulación y bandas de frecuencia diferentes, Pocas probabilidades de ser compatible con estándares anteriores como 802.11b. No soporta voz en la especificación inicial. Costes no fijados, probables que sea relativamente caro.
	HiperLAN/2	54 Mbps/24 Mbps	Respaldado por "grandes nombres", admite la conexión orientada a servicios tales como la transmisión de voz	Probable que sea caro. Competidor directo con 802,11a Probabilidades de ser el perdedor ante la competencia directa.
Home Networking	HomeRF	2 Mbps/1 Mbps; planes futuros soportarán 10 Mbps	Rápida, coste-efectivo a estándares de redes para casa.	Improbable que sea establecido en un entorno fuera de casa.

2.2.7 WiMAX

Es una WAN inalámbrica que pertenece al grupo IEEE 802.16 [Tutorial-Report.com, 2005:07]. Usa microondas que permiten transmisión punto a multipunto. WiMAX tiene un rango de transmisión de unos 50 km, pero se puede usar para la última milla de comunicaciones de banda ancha.

Combinando múltiples canales IEEE 802.16 para una transmisión en un sentido se podría proporcionar anchos de banda por encima de 350 Mbps.

Originalmente, la banda de 10 a 66 Ghz se usa para el estándar IEEE 801.16a que también opera en las licencias de 2 a 11 Ghz. El interés de estas bandas más bajas es que las señales pueden penetrar paredes y obstáculos no metálicos sin necesidad de línea directa.

Este nuevo estándar es compatible con otros anteriores como Wi-Fi (802.11).

2.2.8 ZigBee

En octubre de 2006, Nokia presentó ZigBee, [Tutorial-Report.com, 2005:07] como sustituto de Bluetooth. Es muy similar a dicha tecnología y se rige por el estándar de redes inalámbricas de área personal (WPAN's) IEEE 802.15.4, aunque con un menor consumo de energía.

2.3 MÉTRICAS DE LA SEÑAL

A continuación se van a explicar los diferentes métodos para extraer la información de la señal del sistema de comunicación inalámbrica [Pahlavan, 2002]. Las técnicas y algoritmos de localización procesarán esta información para obtener la localización del dispositivo móvil desconocido.

Las señales del sistema de localización pueden tomar diferentes caminos desde un transmisor a un receptor debido a que inciden en obstáculos, como muebles y paredes, que se encuentran en el escenario. A este fenómeno se le conoce como multipath [Systems, 2005][Li, 2000]. Se puede describir por:

-Line of Sight (LOS): Conexión directa entre el transmisor y el receptor.

-Non-Line-Of- Sight (NLOS): Camino resultante de la señal que llega después de la reflexión en reflectores.

El multipath puede influir negativamente en la extracción de la información, porque produce fluctuaciones en la fase, amplitud, ángulo de llegada y potencia de la señal recibida. Estos parámetros son los que se intentan averiguar mediante las diferentes métricas que se van a estudiar en los párrafos siguientes. Por lo tanto, si los componentes de la señal se modifican durante el camino, la información no será correcta y como consecuencia el resultado de la ubicación será erróneo.

2.3.1 AOA (Angle Of Arrival)

2.3.1.1 Definición

AOA es una técnica de localización que se basa en encontrar la dirección de máxima intensidad de la señal para cada par de antena-dispositivo. AOA se define como el ángulo entre la dirección de propagación de una onda incidente y algunas direcciones de referencia las cuales se conoce como orientación. Orientación, se define como las direcciones fijas contra las cuales se miden los AOA. Se representan en grados tomando como dirección para sus medidas las de las agujas del reloj desde el Norte. Cuando la orientación es 0° o posicionada en el Norte, el AOA es absoluto, si no, es relativo.

2.3.1.2 Método clásico de estimación de AOA

Una forma común para obtener medidas de AOA [Maurício Henrique Costa Dias, 2005] consiste en usar un array de antenas en cada nodo sensor. Se asume para el problema de la estimación de AOA que se trabaja en campo lejano, donde las ondas son planas y el array está suficientemente lejos del origen.

Las condiciones son:

$$\begin{aligned}r &> 2 \times \frac{D^2}{\lambda} \\r &\gg \lambda \\r &\gg D\end{aligned}$$

r es la separación entre el array y el origen
 D es la máxima dimensión del array (anchura o altura)
 λ es la longitud de onda de la señal

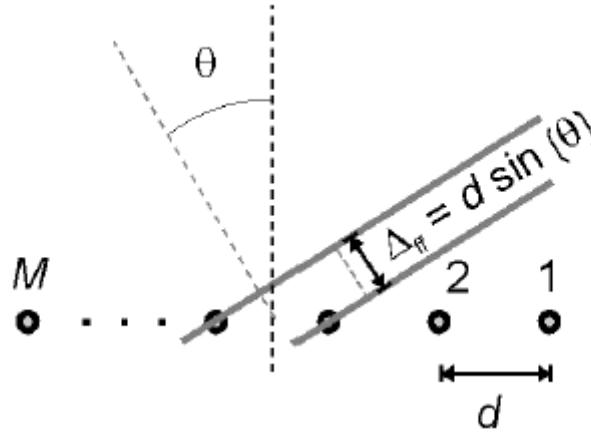


Figura 6: Modelo de estimación de AOA.

Si una señal $s(t)$ incide a M -antenas ULA en un ángulo AOA θ , $y(t)$ es un vector que se forma a la salida de la antena de la siguiente forma:

$$\begin{aligned} y(t) &= a(\theta) \times s(t) + n(t) \\ y(t) &= [y_1(t) \dots y_M(t)]^T \\ n(t) &= [n_1(t) \dots n_M(t)]^T \end{aligned}$$

donde T es la transposición de un vector o matriz, $y_i(t)$ es la señal de la antena i^{th} y $n_i(t)$ es el ruido de la antena i^{th} , normalmente considerado como Gaussiana.

A $a(\theta)$ se conoce como el vector de manejo, contiene la información del ángulo de llegada y se da por:

$$a(\theta) = [1 \times \exp^{-jWcR_2} \dots \exp^{-jWcRM}]^T$$

$$R_k = \frac{(k-1) \times d \times \sin \theta}{v} = \frac{(k-1) \times \Delta_{ff}}{v}$$

donde Wc es la frecuencia de la señal, R_k es el TDOA (tiempo de diferencia de llegada) del campo lejano entre k_{th} y la primera antena, D es el espacio entre un par de antenas, Δ_{ff} es el campo lejano de separación entre las ondas que inciden en antenas consecutivas y V es la fase de la señal que incide.

2.3.1.3 Errores de la estimación de la dirección

Los errores de la estimación [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] de la dirección dependen de la influencia del multipath sobre la señal, la posibilidad de tener visión directa entre el transmisor y receptor y la exactitud del ángulo del array de antenas.

2.3.1.4 Exactitud de la técnica

La técnica será eficiente en función del entorno, el canal por donde se propaga la señal, el algoritmo usado para la estimación de la dirección de llegada, la exactitud del ángulo del array y de antenas, el número de estaciones fijas que se han utilizado y la localización geográfica del dispositivo móvil relativo al AP.

2.3.2 TOA (Time of arrival)

2.3.2.1 Definición

El sistema TOA mide la distancia entre un transmisor y un receptor (entre estaciones móviles y/o diferentes AP's) en un espacio libre, basado en estimar el retardo de propagación de las señales que viajan a la velocidad de la luz [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001].

2.3.2.2 Método estimación de TOA

El TOA puede ser hallado midiendo:

- La fase de la señal portador de banda estrecha recibida.
- El tiempo de llegada de un pulso de ancho de banda estrecho.

A continuación se divide esta técnica en cuatro clases:

1. Señales de banda estrecha y Sistemas de medidas de fase

En las técnicas de banda estrecha la diferencia de fase entre las señales portadoras recibidas y transmitidas se usa para medir la distancia entre dos puntos. La fase de una señal portadora recibida, Φ , y el TOA de la señal, r , se relacionan de la siguiente forma:

$$r = \frac{\Phi}{W_c}$$

donde W_c es la frecuencia de la portadora en radianes.

2. Señales de banda ancha y Técnicas de Superresolución

Un transmisor transmite una señal cifrada por una secuencia de pseudo-ruido (PN) conocida. Después un receptor correlaciona (usa correlación cruzada) la señal recibida con una secuencia PN, que se ha generado en el propio receptor, usando correlación o filtros adaptados. La distancia entre el transmisor y receptor se determina con el tiempo de llegada del pico del primer correlador. Un ejemplo de técnicas de Superresolución es MUSIC.

3. Técnicas UWB

El ancho de banda de la señal es uno de los factores que afectan a la exactitud de la estimación de TOA en entornos multipath. Cuanto más grande sea el ancho de banda, mayor es el rango de exactitud. Los sistemas UWB, que trabajan en anchos de banda de 1 Ghz, son adecuados para medir la exactitud TOA para aplicaciones de entornos indoor.

La forma de acercarse al instante en el que la salida del filtro logra su pico, proporciona la estimación del tiempo de llegada. Una estimación óptima del tiempo de llegada se obtiene usando un filtro adaptado o un banco de correladores receptores. Estas dos técnicas son matemáticamente equivalentes, se elige en función del coste.

La estimación de la máxima verosimilitud (MLE) o probabilidad a posteriori (MAP) del tiempo de llegada también se puede reducir con la estimación basada en filtros adaptados o receptores correladores, cuando el canal de comunicación se pueda modelar como un canal AWGN.

En vez de estimar la posición óptima por MLE / MAP, el esquema TOA convencional estima la localización del nodo usando la técnica de mínimos cuadrados (LS):

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^N w_i \left[\frac{\tau_i - d_i(\theta)}{c} \right]^2$$

Donde N es el número de nodos de referencia, τ_i es la medida i th TOA, $d_i(\theta) := \|\theta - \theta_i\|$ es la distancia entre el nodo dado y el nodo de referencia i th, con θ y θ_i denotando sus respectivas localizaciones, y w_i es el factor escalar de peso para la medida i th que refleja la fiabilidad de la estimación i th TOA.

4. Triangulación

Para obtener un resultado aceptable con esta técnica, es necesario realizar medidas de tiempo de al menos tres AP's [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001]. Las medidas permiten trazar circunferencias con centro en cada AP. La intersección de las tres circunferencias da un punto que se identifica con la localización de la estación móvil. TOA requiere sincronización entre los transmisores y receptores.

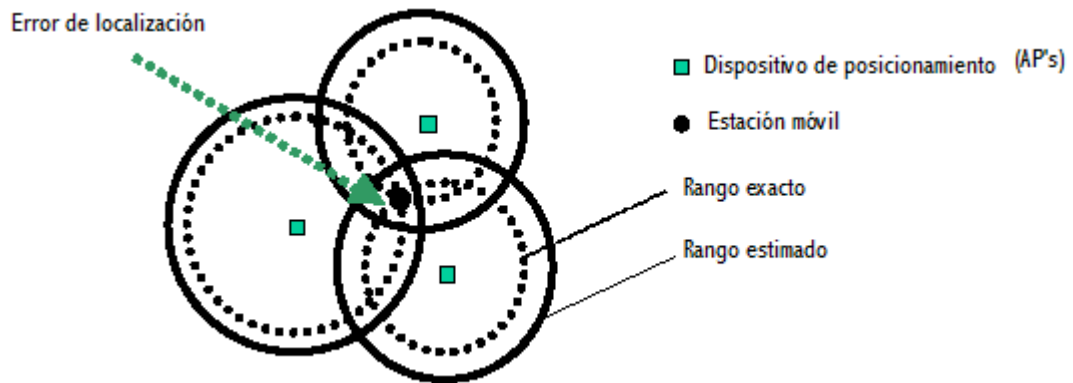


Figura 7: Método de triangulación TOA.

2.3.2.3 Errores de la estimación de la dirección

Los errores de la estimación de la dirección depende de:

- Multipath
 - Filtros adaptados y correladores:

En filtros adaptados o correladores, el tiempo en el cual la salida del filtro hace picos, o, la diferencia del tiempo de la señal modelo que produce la máxima correlación con la señal recibida, se usan como el tiempo estimado. En un sistema de banda estrecha, sin embargo, este valor no debe ser el verdadero TOA debido a las múltiples réplicas de la señal transmitida, debido parcialmente al solape y el cambio de posición de los picos de correlación. En otras palabras, los canales multipath crean relaciones erróneas entre la señal recibida y la señal modelo. Como resultado, en lugar de la auto-correlación, se obtiene una correlación cruzada que no nos da el pico que corresponde con el tiempo correcto.

- Señales de banda estrecha y sistemas de medidas de fase

Cuando una señal portadora de banda estrecha se transmite en un entorno multipath, la composición de la señal portadora recibida es la suma de un número de portadoras, que llegan a lo largo de diferentes caminos, de la misma frecuencia pero diferente amplitud y fase.

Por este motivo, esta técnica no nos proporciona estimaciones exactas.

- Non Line of Sight (NLOS)
- Exactitud en la sincronización entre transmisor y receptor.

2.3.2.4 Exactitud de la técnica

La fiabilidad de la técnica depende [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] del entorno indoor, el algoritmo que se use para estimar el tiempo, el número de estaciones fijas (AP's) usadas y la ubicación del dispositivo móvil con respecto a la posición de la estación fija.

2.3.3 TDOA (Time difference of arrival)

2.3.3.1 Definición

El sistema TDOA [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] emplea la diferencia entre los tiempos de llegada de la señal entre la estación móvil y diferentes AP's.

2.3.3.2 Método estimación TDOA

Los métodos de estimación de TDOA [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] son los mismos que se han comentado en el punto de Método estimación de TOA, con la diferencia que en el método de Triangulación, la figura geométrica que se dibuja es la hipérbola, ya que esta es la curva que satisface la condición que su distancia a dos referencias sea una constante. El punto donde se corten las hipérbolas es donde se encuentra la estación móvil. La medida de error es inversamente proporcional al ancho de banda e independiente de la distancia entre el AP y la estación. Al igual que TOA, requiere sincronización para evitar errores de precisión, pero sólo entre los receptores o AP's.

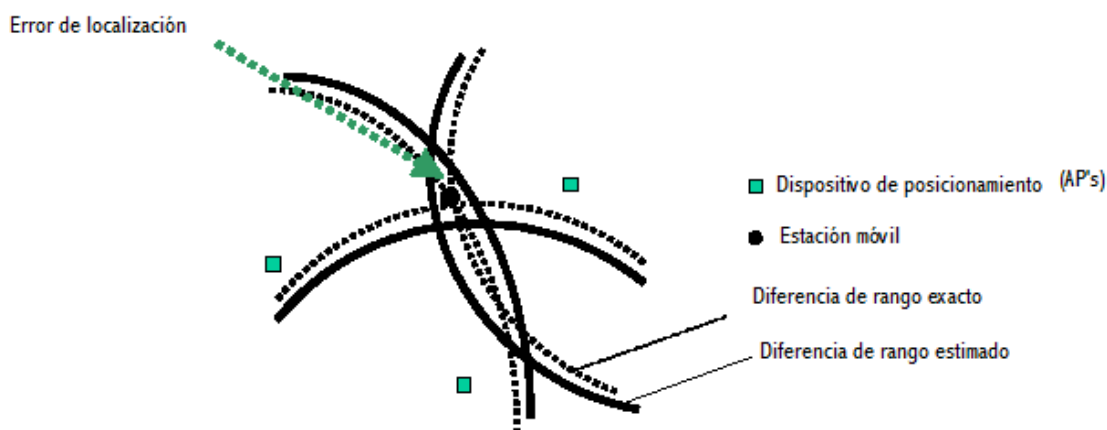


Figura 8: Método de triangulación TDOA.

La ventaja sobre TOA es que no requiere LOS, ya que la diferencia de tiempos cancela errores por reflexiones. Sin embargo, en algunos entornos, como en áreas urbanas, es necesario efectuar las medidas con respecto a cuatro AP's.

2.3.3.3 Exactitud de la técnica

La exactitud de la técnica TDOA [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] depende de las mismas indicaciones que se realizaron en la técnica TOA en el punto 2.3.2.4.

2.3.4 RSSI (Receive Signal Strength Indication)

2.3.4.1 Definición

Es el método que mide la distancia entre los AP's y estaciones móviles usando el nivel de potencia recibido.

2.3.4.2 Método de estimación

En esta técnica, las potencias recibidas de los puntos de referencia inalámbricos, se recogen y posteriormente esta información se utiliza para determinar la localización de un objeto.

La mayor ventaja de ésta técnica es su simplicidad. RSSI tiene una métrica sencilla que se puede medir fácilmente en la mayor parte de los dispositivos móviles. Por ejemplo, la capa MAC de IEEE 802.11 WLAN estándar proporciona información RSSI desde todos los AP's en un cuasi período de señal de trama que puede ser usado como una métrica para la localización.

La técnica RSSI sufre mayor rango de errores comparado con TOA, pero estos errores se pueden reducir incrementando el número de AP's en un determinado entorno indoor.

2.3.4.3 Errores de la estimación de la dirección

Los errores de la estimación dependen [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001] del efecto multipath, NLOS, Local Shadowing y del algoritmo utilizado para resolver las ecuaciones no lineales.

2.3.4.4 Precisión de la técnica

El entorno indoor, el algoritmo usado para estimar la localización del usuario, el número de puntos de acceso y el modelo del camino de pérdidas que se seleccione influyen en la precisión de la técnica RSSI [Chahé Nerguizian, Charles Despins and Sofiéne Affe., 2001].

2.3.5 Cell- Id

Es el método más simple para determinar la posición de una estación móvil. Se asume que la posición de la estación móvil corresponde con la posición del AP que tenga la señal de potencia más fuerte recibida por la estación móvil.

2.3.6 Comparación entre TOA, AOA, RSSI y Cell-Id

Los canales radio indoor [Ahmad Hatami] sufren la propagación de caminos multipath y fuertes caídas, así que las medidas de RSSI y AOA proporcionan métricas menos exactas que TOA.

En estos mismos entornos, debido a la obstrucción de paredes, techos, u otros objetos, el camino de propagación DLOS no es siempre el que ofrece una potencia (RSSI) en algunos casos como por ejemplos NLOS.

Para estimar exactamente el TOA en áreas indoor, se necesitan técnicas de procesamiento. Los sistemas basados en TOA son sensibles a la disponibilidad del ancho de banda y también a la posibilidad de no detectar el camino directo. Sin embargo los sistemas basados en RSS son menos sensibles a la disponibilidad del ancho de banda y más resistentes a las condiciones de la presencia del camino directo pero su exactitud se degrada ante la presencia de multipath. El tiempo requerido para la localización es menos importante. A pesar de ello, los sistemas basados en TOA pueden proporcionar mayor exactitud que RSSI en presencia del camino directo dominante que no es común en entornos indoor.

También existe un inconveniente en la utilización de RSS en los sistemas de posicionamiento: las localizaciones de los equipos móviles hacen referencia a los puntos que forman un mapa-radio que depende del entorno, como consecuencia los sistemas RSSI son restrictivos a un área determinada y no escala correctamente en grandes áreas.

Tabla 1: Comparación de técnicas de localización.

Técnicas	Degradación de la estimación		Complejidad de los AP's
	Multipath	Distancia (AP-STA)	
TOA/TDOA	Ligeramente	Ninguno	Medio
AOA	Ligeramente	Ligeramente	Alta
RSSI	Severamente	Ligeramente	Baja
CELL-ID	Ninguno	Severamente	Baja

En esta tabla [Yamasaki, 2005] se encuentra recogida la influencia del multipath y de la distancia entre el punto de acceso y la estación móvil para las diferentes técnicas que se han estudiado. También indica la complejidad de los AP's en función de la métrica elegida.

2.5 CLASIFICACIÓN DE LOS SISTEMAS DE LOCALIZACIÓN

Los sistemas de localización están basados en técnicas que permiten obtener y procesar la información que proporcionan las métricas elegidas (AOA, TOA, TDOA o RSSI) para conseguir la ubicación de un terminal móvil.

En este proyecto se explican dos de las técnicas que se utilizan para dicho fin, que son triangulación y Fingerprint.

2.5.1 Triangulación

2.5.1.1 Definición

La triangulación [*Wikipedia*] es un proceso por el cual la localización de un transmisor radio puede ser determinado por medidas de cada uno de las distancias radiales, o la dirección de la señal del receptor de tres o cuatro puntos diferentes.

2.5.1.2 Técnica de triangulación

La técnica de detección de localización de triangulación [*Hightower & Borriello, 2001*] usa la propiedad de la geometría de triángulos para computar localizaciones de objetos.

La triangulación se divide en dos categorías:

- **Lateración (*Lateration*)**

Lateracion calcula la posición de un objeto midiendo su distancia a múltiples posiciones de referencia. Para calcular la posición de un objeto en dos dimensiones requiere medidas de distancias de tres puntos no colineales. En tres dimensiones, se requieren medidas de distancias desde cuatro puntos no colineales.

Hay tres aproximaciones para mediciones de distancias requeridas para técnicas de lateración.

- 1.- Directo

Son difíciles de obtener automáticamente debido a la complejidad de coordinar los movimientos autónomos físicos.

- 2.- Tiempo de vuelo

Las medidas de distancia de un objeto a algún punto P usando el tiempo de vuelo consiste en medir el tiempo que tarda en viajar una señal entre el objeto y un punto P a una velocidad conocida. El objeto se debe mover como un avión viajando a una velocidad conocida para un intervalo de tiempo, o que el objeto esté prácticamente estacionario y se intente observar la diferencia en transmisión y tiempo de llegada de una señal emitida.

- 3.- Atenuación

La intensidad de una señal emitida disminuye a medida que aumenta la distancia de la fuente de emisión. La disminución de la intensidad original es la atenuación. Dada una función que correlaciona con la atenuación y la distancia para un tipo de emisión y la potencia original de la emisión, es posible estimar la distancia de un objeto a algún punto P por medidas de potencia de la emisión cuando alcance un punto P. Por ejemplo, una señal de radio en espacio libre emitida por un objeto será atenuado por un factor proporcional a $1/r^2$ cuando alcanza el punto P a una distancia r de un objeto.

En entornos con muchas obstrucciones como en una oficina, las medidas de distancia usando atenuaciones son normalmente menos exactas que el tiempo de vuelo. La señal

de propagación debido a las reflexiones, refracciones y multipath causa la atenuación para correlar pobremente, resultando que la distancia que se estima sea inexacta e imprecisa.

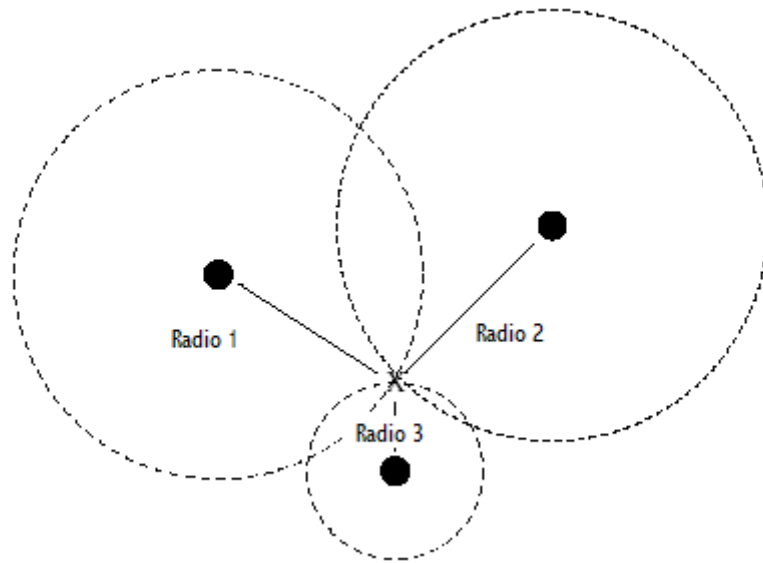


Figura 9: Técnica de lateración.

La figura muestra como se determina una posición en 2D usando lateración. La lateración requiere medidas de distancias entre el objeto denominado "X" y tres puntos no colindantes.

- **Angulación (Angulation)**

La angulación es similar a la lateración, excepto que en lugar de distancias, se usan ángulos para determinar la posición de un objeto. En general, dos dimensiones de angulación requieren dos medidas de ángulos y una longitud así como la distancia entre los puntos de referencia de ángulos para especificar la posición precisa. La fase de los array de antenas es idónea para la técnica de angulación. Dada la diferencia en tiempos de llegada y la geometría del array de recepción, es también posible calcular el ángulo del de la emisión original. Si existen suficientes elementos en el array y suficiente separación, se puede calcular la angulación.

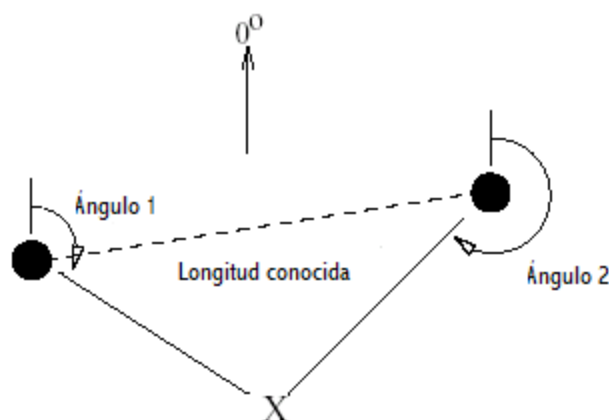


Figura 10: Técnica de angulación.

Este ejemplo de angulación de 2D ilustra la localización de un objeto “X” usando ángulos relativos a un vector de referencia 0° y la distancia entre dos puntos de referencia. La angulación de 2D siempre requiere al menos dos ángulos y una medida de distancia para localizar la localización del objeto.

2.5.2 Técnica Fingerprint

Es una técnica de localización [Patmanathan, 2006:08] que parte del hecho que las características de la señal de propagación son diferentes en cada punto de localización de la zona de interés. De esta forma cada punto de localización tiene un único Fingerprint.

2.5.2.1 Definición de Fingerprint

Un Fingerprint es un vector de valores del mismo tipo de métricas de la señal que corresponde a una posición dada. El tamaño del Fingerprint depende del número de sensores o puntos de acceso (AP) que se puedan escuchar, es decir, que simultáneamente se encuentre dentro de las regiones de cobertura de los AP's.

2.5.2.2 Metodología de la técnica Fingerprint

Esta técnica se divide en dos partes:

- Fase Off-line

También denominada de aprendizaje o calibración. Consiste en coleccionar un conjunto de datos en forma vector. La información está constituida por parámetros de la señal que se han medido al determinar la locación, ya conocida previamente, de un usuario. Los AP's son los que emiten la información (AOA, TOA, TDOA o RSSI). Cada Fingerprint se recoge en función de un período de tiempo.

En ocasiones, cuando los cambios del entorno son importantes, como cambios en el mobiliario, paredes, recolocación de AP's) se necesita realizar una re-calibración, ya que las señales de propagación se van afectadas por estas modificaciones y así mismo la información Fingerprint, ya no sería veraz. La representación gráfica del conjunto de Fingerprint constituye un mapa del entorno.

- Fase Real – time

Un usuario se crea un vector formado por los parámetros de la señal que recoge de las zonas de cobertura de los AP's que alcanza dentro del entorno. Este vector se compara con todos los Fingerprint que forman el conjunto de información mediante un algoritmo que estima la localización óptima del nuevo usuario.

2.5.2.3 Predicción Fingerprint

Las señales se ven afectadas por las estructuras (paredes suelos, techos...) mobiliario y materiales usados que existan en un entorno indoor, ya que distorsionan sus parámetros característicos y como consecuencia repercute en la exactitud de la localización [Widyawan]. Por este motivo se pueden usar modelos de propagación que ayuden a predecir la información Fingerprint.

Hay dos tipos de modelos: empíricos y deterministas. Las ventajas del empírico es la velocidad, ya que sólo se necesita una entrada y las fórmulas que se aplican son sencillas. Las desventajas más significativas son la poca exactitud y la incapacidad para predecir los parámetros de ancho de banda del canal de comunicación. Los modelos deterministas intentan seguir los principios físicos de la propagación de ondas electromagnéticas. Las más populares son ray tracing y ray launching. Están basados en principios ópticos geométricos.

Estos modelos son muy exactos y pueden predecir parámetros de ancho de banda de paredes. Por otro lado son lentos, la velocidad se puede aumentar implementando algún pre-proceso o aplicando

simplificaciones. También hay modelos semi-deterministas que emergen para combinar aproximaciones empíricas y deterministas y ganar sus ventajas.

Algunos modelos a destacar son:

- One – Slope Model

El One-Slope Model (ISM) es la forma más sencilla para procesar el nivel medio de la señal dentro de un entorno indoor sin conocimiento de los detalles del mapa del entorno. El camino de pérdidas es justo una función de distancia entre la antena transmisora y receptora.

$$L(d) = L_0 + 10n \log(d)$$

Donde L_0 (dB) es una referencia del valor de pérdidas para la distancia de 1 m, n es un factor de pérdida de potencia y d (m) es una distancia. L_0 y n son parámetros empíricos para un entorno dado que controlan la predicción.

- Multi-wall Model

El Multi-Wall Model (MWM) lleva factores de penetración en la pared y suelo junto con las pérdidas de espacio libre. Los factores de pérdidas de transmisión de las paredes o suelos L_{paredes} o L_{suelos} respectivamente, más la suma de la distancia en línea recta de dos antenas, da el factor de penetración de pérdidas totales.

Esta señal de pérdidas se da por:

$$L_{\text{MWM}}: L_1 + 20 \log(d) + L_{\text{paredes}} + L_{\text{suelos}}$$

Donde

$$L_{\text{paredes}} = \sum a_{wi} k_{wi} \quad i = 1, I$$

$$L_{\text{suelos}} = a_f K_f$$

donde L_{MWM} denota la pérdida de la señal estimada (dB); L_1 es el espacio libre de pérdidas a distancia de 1 m. del transmisor (dB), L_{paredes} es la contribución de las paredes a la señal de pérdidas total (dB), L_{suelos} es la contribución de los suelos a la señal total de pérdidas (dB), a_{wi} es el factor de pérdidas de transmisión de una pared de tipo i_{th} (dB), k_{wi} es el número de paredes de tipo i_{th} , I es el número de tipos de paredes, a_f es la transmisión del factor de pérdidas de un suelo (dB), K_f es el número de pisos.

- Motif Model (MM)

El MM es una combinación de ventajas de las aproximaciones empíricas y determinísticas. El modelo se basa en una técnica modificada de ray-launching. La propagación Ray toma ventajas de la técnica modificada line-drawing que divide el entorno en una rejilla. Un simple mapa de bits de un plano de una planta sirve como la base de los datos de entrada. Los píxeles fijados representan paredes, particiones y obstáculos. Diferentes colores distinguen diferentes materiales. La predicción de la propagación se calcula en todos los elementos vacíos. La potencia de la señal se estima en cada elemento vacío de un mapa de bits grabando el número de radios (rays) que pasan por ahí.

2.6 ALGORITMOS DE UN SISTEMA DE LOCALIZACIÓN

A continuación se van a explicar un conjunto de algoritmos que se han empleado para procesar la información que se ha obtenido como resultado de la ejecución de las técnicas de triangulación y

Fingerprint para determinar y/u optimizar una localización. Los algoritmos a destacar son la Teoría Bayesiana, las Redes neuronales y k-NN.

2.6.1 Métodos heurísticos: Teoría Bayesiana

2.6.1.1 Definición

Se basa en la aplicación de la Teoría de Bayes [Fernandez, 2007]:

$$P\left(\frac{h}{D}\right) = \frac{P\left(\frac{h}{D}\right) \times p(h)}{P(D)}$$

El objetivo del sistema de localización Bayesiano consiste en estimar la función de densidad de la posición x_t en un instante t , a partir de las observaciones tomadas con anterioridad y la probabilidad a priori $P(x_1)$.

Para describir la localización Bayesiana, se considera que se tiene una secuencia de variables aleatorias que representan la localización de un individuo en sucesivos instantes de tiempo:

$$(x_1, x_2, x_3, x_4, \dots, x_t)$$

donde x_t está formado por los valores que determinan la localización de un individuo (como coordenadas cartesianas, θ , zonas....).

En cada instante de tiempo, se obtiene una serie de observaciones, que corresponde con los valores que toman los atributos definidos en el problema en t (como la distancia, potencias, velocidad, incrementos de posición, a , z)

- $P(D) = P(a_1, a_2, \dots, z_1, z_2)$ Probabilidad a priori de que los atributos tomen un valor determinado. Representa los valores de los atributos de la nueva instancia a clasificar.
- $P\left(\frac{x_t}{a_1, a_2, \dots, z_1, z_2}\right)$ Probabilidad a posteriori de que los atributos tomen unos determinados valores, sabiendo que el usuario se encuentra en una zona determinada.
- $P\left(\frac{x_t}{a_1, a_2, \dots, z_1, z_2}\right)$ Probabilidad a posteriori de que el usuario se encuentre en una determinada posición, una vez conocidos los valores de los atributos.

Este son los datos que determinan la clasificación de la nueva instancia.

La localización del usuario es aquella función a posteriori $P\left(\frac{x_t}{a_1, a_2, \dots, z_1, z_2}\right)$ que dé una mayor probabilidad.

$$P\left(\frac{a_1, a_2, \dots, z_1, z_2}{x_1}\right) \times \frac{P(x_1)}{P(a_1, a_2, \dots, z_1, z_2)} > \text{ó} < P\left(\frac{a_1, a_2, \dots, z_1, z_2}{x_2}\right) \times \frac{P(x_2)}{P(a_1, a_2, \dots, z_1, z_2)}$$

$$x_i \quad i \in [1, N]$$

Para poder aplicar este método hay que partir de dos hipótesis:

1. Condición de Markov
Se considera sobre el modelo dinámico que la localización actual de un usuario sólo depende de la localización y de los valores que tomaron los atributos en el instante inmediatamente anterior.
2. Independencia sobre las observaciones o atributos, para facilitar el cálculo de las probabilidades.
De esta forma la probabilidad conjunta es:

$$\frac{P(x_t)}{P(a_1, a_2, \dots, z_1, z_2)} = P\left(\frac{x_t}{a_i}\right) \times P\left(\frac{x_t}{z_i}\right)$$

2.6.1.2 Técnicas de localización Bayesiana

Existen diferentes modelos de localización Bayesiana, en función de las técnicas propuestas para la estimación computacional de la densidad de probabilidad a posteriori con $P(x)$ como una función normal. Sus parámetros se estiman mediante:

a. Filtro de Kalman

Es un conjunto de ecuaciones matemáticas [López, 1999] que permite estimar la posición de un usuario mediante mínimo cuadrados, en un instante de tiempo t , a partir de la actualización de la información disponible en $t-1$. La actualización se realiza con los datos que se obtienen en t .

El filtro de Kalman, se puede definir también como un algoritmo de pronóstico (1) y de corrección (2) donde en (1) se hace una estimación del estado futuro con la información disponible en ese momento. Y en (2) se hace una mejora del resultado de (1) minimizando el error estadísticamente.

b. Rejillas de probabilidad:

Se discretiza todo el posible espacio de localizaciones del usuario y se calcula en cada intervalo de discretización una función de densidad a posteriori constante.

c. Métodos topológicos.

El entorno se representa mediante nodos que identifican partes significativas del área de localización, como final de pasillo, puertas,.....

Cada nodo tiene unos estados a definir, como avance o retroceso.

En función del grado de discretización del entorno de localización, es decir, el número de nodos que representan diferentes zonas dentro del entorno, mejorará la carga computacional, en decremento de la precisión de la estimación de la posición del usuario.

2.6.1.3 Problemas

Para aplicar el método Bayesiano, es necesario:

1. Disponer de un gran número de ejemplos de entrenamiento.
2. Se necesita conocer alguna información adicional, como las funciones de densidad de probabilidad de los atributos.

En caso contrario, sería muy complicado calcular las probabilidades que intervienen en el teorema de Bayes.

2.6.2 Métodos heurísticos: Redes neuronales

2.6.2.1 Definición

Una red neuronal artificial (RNA) [Cáceres, 2003] es un sistema que simula el cerebro humano. Es un modelo de procesamiento de la información que requiere aprender a comportarse (learning) y entrenarse (training) a partir de un conocimiento previo del entorno del problema.

El sistema está formado por elementos distribuidos, llamados neuronas, agrupados en capas (layers), y que se encuentran interconectados (sinapsis). Esta estructura suele tener varias entradas y salidas.

Reúne las características de adaptabilidad, robustez, capacidad de generalización y tolerancia a fallos.

2.6.2.3 Estructura de la neurona

La neurona o unidad de procesamiento tiene la estructura mostrada en la siguiente figura:

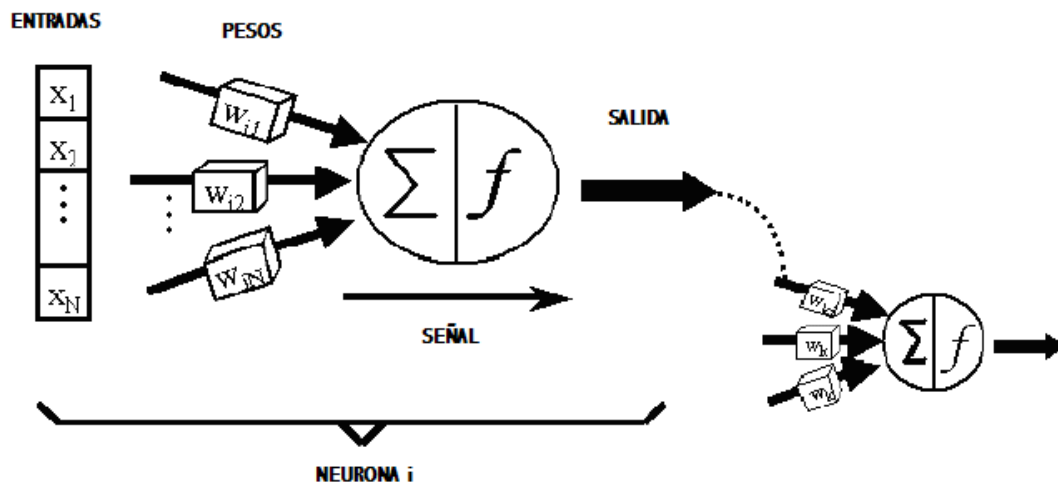


Figura 11: Modelo de Mc Culloch- Pitts

Como se puede observar, la salida de una neurona suele ser una de las entradas del siguiente elemento. Las N señales de entrada están representadas por x_1, x_2, \dots, x_N . Están asociadas a los pesos, $w_{i1}, w_{i2}, \dots, w_{iN}$, los cuales determinan el nivel de influencia de la neurona j para la neurona i .

Cada neurona se subdivide en dos etapas: suma y activación.

1. SUMA:

Las entradas y los pesos se combinan de la siguiente forma:

$$y_i = \sum_{j=1}^N w_{ij} x_j$$

y_i , que representa el estado interno de la neurona i , se le aplica en la etapa de activación.

2. ACTIVACIÓN:

La función que se aplica $x_i = f(y_i)$ denominada de activación, limita la salida a $[-1, 1]$ o $[0, 1]$, o en el caso de que x_i sea discreto $\{-1, 1\}$ o $\{0, 1\}$.

2.6.2.4 Arquitectura de red

La red neuronal está formada por capas, las cuales se clasifican en:

- Capa de entrada
- Capas intermedias u ocultas
- Capa de salida

En función de las conexiones entre las neuronas se puede diferenciar: FeedForward y Recurrentes.

- FeedForward

Se caracterizan porque la salida de una neurona es la entrada del resto de elementos que forman la siguiente capa. Un ejemplo es el perceptrón multicapa (Multilayer Perceptron, MLP).

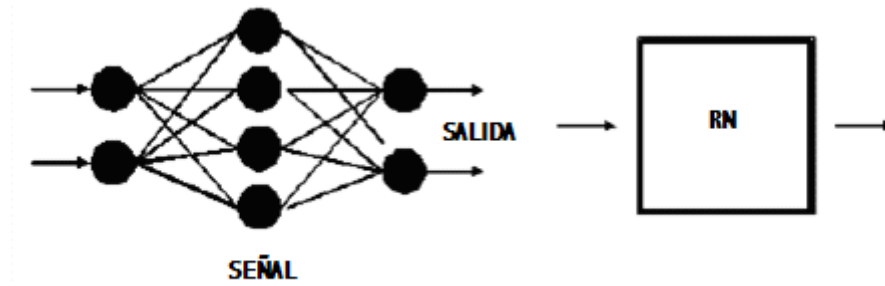


Figura 12: Red neuronal FeedForward.

La arquitectura de perceptrón multicapa se organiza de la siguiente forma: Las señales fluyen secuencialmente a través de diferentes capas de la entrada a la capa de salida. Para cada capa, cada neurona primero calcula un producto escalar entre un vector de pesos y el vector dado por las salidas de las capas previas (Subcapa suma). Posteriormente al resultado se aplica la función correspondiente a la subcapa de activación, para producir la entrada a la siguiente capa.

- Recurrentes

Son aquellas que poseen conexiones de realimentación. Un ejemplo es el modelo de Hopfield.

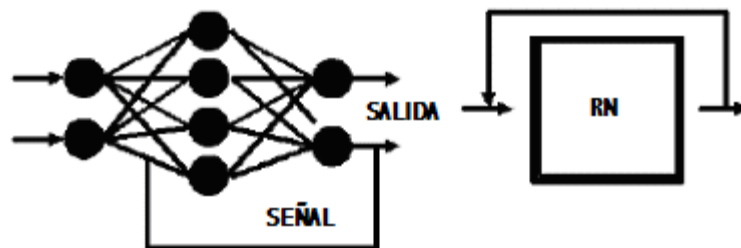


Figura 13: Red neuronal recurrente.

2.6.2.5 Algoritmos de aprendizaje (entrenamiento)

Una de las características de redes neuronales es la capacidad de aprendizaje, que consiste en modificar el valor de sus pesos a partir de datos etiquetados, para que el sistema dé una respuesta óptima ante una nueva entrada.

Algunos algoritmos de aprendizaje son:

- Aprendizaje por corrección de errores

Se basa en la regla Delta, para minimizar la función de error buscando un gradiente descendente.

- Aprendizaje competitivo

Dos neuronas entre si compiten para permanecer activas y de esta forma participar en el proceso de aprendizaje.

- Aprendizaje Hebbiano

Consiste en dos neuronas que están activas con conexiones entre sí.

- Aprendizaje de Boltzman

Consiste en ajustar los pasos de conexión, para que el estado de los elementos dibuje una determinada distribución deseada.

Otro tipo de clasificación viene determinada por la relación que tienen los elementos de la red neuronal con los datos de entrenamiento:

- Aprendizaje supervisado

Un agente externo indica a la red la respuesta deseada para una entrada etiquetada.

- Refuerzo

La información que le ofrece a la red es una corrección a la salida, no la respuesta.

- Aprendizaje no supervisado

No existe un agente externo que le dé información sobre la respuesta óptima ante un dato etiquetado.

Es esencial notar que el objetivo del algoritmo de aprendizaje es construir un modelo con buena capacidad de generalización cuando compara con nuevos valores de entrada, valores no presentes en el conjunto de entrenamiento. La generalización se relaciona con el número de parámetros y de datos etiquetados y la duración del entrenamiento. Cuantos más datos etiquetados estén disponibles y más larga sea la fase de training, menor será el error pero también disminuye la capacidad de generalización, ya que el sistema memoriza los patrones de entrenamiento y no extrae las regularidades que hace posible llegar a la generalización. La teoría de la dimensión de Vapnik-Chervonenkis (VC) [Roberto Battiti, 2002] es apropiada para conseguir generalización en el algoritmo, pero la dimensión VC no se calcula fácilmente para un problema específico. Así que, como alternativa, se propone experimentar con el número de parámetros y la duración de la fase de entrenamiento para que el algoritmo llegue a generalizar de forma óptima.

Los modelos de redes neuronales y técnicas de aprendizaje automático son una solución eficiente para estimar la localización de un objeto y/o individuo [Uzair Ahmad1] [Roberto Battiti, 2002]. En este caso, los valores de la capa de entrada serán los atributos de los AP's que el objeto ha detectado (potencia, coordenadas) y la salida será la localización óptima de dicho objeto.

A continuación se intenta explicar un ejemplo de localización utilizando una red perceptrón multicapas.

Como se ha indicado en la definición de este algoritmo, el proceso se divide en dos fases, entrenamiento y clasificación. La primera fase consiste en averiguar los pesos adecuados a partir de muestras etiquetadas, para obtener la menor probabilidad de error. Esto se traduce en localizaciones exitosas ante nuevos datos. Y la segunda fase, consiste en obtener la localización de un usuario desconocido a partir de los pesos hallados en la etapa de entrenamiento.

- Fase de aprendizaje

Se aplica la función suma a cada uno de los vectores de entrada de la neurona, formados por los atributos de la señal, cuyo resultado se utilizará para variar el valor del peso. Se realizarán tantas iteraciones como muestras etiquetadas se dispongan.

Posteriormente se aplica la función de activación, que ofrece como resultado la entrada a la siguiente capa de perceptrones.

Este mecanismo se repite en cada capa de neuronas, hasta llegar a la última, donde su salida será la localización del vector entrada. En este punto, también se calcula el error por mínimos cuadrados (SSE) entre la ubicación obtenida y la conocida previamente.

La secuencia de pasos que se ha comentado, se inicia de nuevo hasta alcanzar un SSE que asegure una ubicación óptima ante nuevas entradas.

- Fase de clasificación.

Se aplica a una nueva entrada la función suma y activación pero con los valores de los pesos que han proporcionado un mínimo SSE en la etapa que se ha denominado como de aprendizaje.

2.6.3 Métodos heurísticos: k vecinos más cercanos

2.6.3.1 Definición

K- Nearest Neighbour [Teknomo, 2006] [Montero] es un algoritmo de coincidencia de patrones (pattern matching). El objetivo de este algoritmo consiste en clasificar un nuevo usuario a partir de otros de su misma clase, que serán aquellos que tengan las características más similares al objeto a clasificar.

2.6.3.2 Algoritmo KNN

El algoritmo KNN está constituido por dos fases:

1º Entrenamiento

Consiste en construir un set o conjunto de datos. Estos datos están formados normalmente por unos atributos y una etiqueta que determina a qué clase pertenecen.

2º Clasificación

Determina ante un nuevo dato o consulta a qué clase pertenece.

El algoritmo KNN realiza la clasificación basándose en la mínima distancia entre una nueva consulta y cada uno de los datos que forman el conjunto de datos de aprendizaje. Los k vecinos más cercanos serán los que correspondan con las k mínimas distancias obtenidas.

KNN puede trabajar con datos cuyas variables o atributos sean no solo cuantitativos, si no, también ordinales y nominales. Si los datos son cuantitativos la distancia se corresponde con la distancia entre coordenadas que forman una dimensión de tantos planos como atributos haya. Y el valor final del nuevo dato se obtendría tomando cualquier media geométrica de los k-vecinos más cercanos.

En el caso contrario, habría que calcular la distancia para datos multivariantes, y una vez obtenidos los k-vecinos más cercanos, éstos se analizarían para determinar que rasgo o etiqueta o variable de similitud se ha repetido más entre ellos.

La mayor parte de los datos reales, consisten en una mezcla de variables de diferentes naturalezas, es decir, nominales, ordinales y cuantitativas coexistiendo en un mismo caso. Para manejar esta situación se introduce el término peso, que mide la contribución de cada variable acorde con la distancia a la consulta, y dando mayor peso (w_{ij}) al K_{th} vecino más cercano.

Los pasos para introducir esta variación al algoritmo son:

1. Usar las distancias normalizadas o similares para todas las variables.
2. Determinar el peso de cada variable.
3. Agregar el peso:

$$d_{ij} = \sum w_{ijk} \times \frac{f_{ijk}}{\sum w_{ijk}} \quad k = 1, n$$

donde f_{ijk} representa la no-similitud entre los datos i y j, que en el caso que se estudia se determina por la distancia.

Los pesos se pueden determinar basándose en un proceso previo de calibración con los datos de entrenamiento que consiste en minimizar el error cuadrático medio (MMSE) o bien como:

$$w_{ij} = \frac{1}{d(x_i, x_j)^2}$$

x_i = nueva consulta

x_j = dato del conjunto de entrenamiento

Si se usa el peso, da la sensación de usar todos los ejemplos del entrenamiento para determinar el valor de la consulta, no sólo los k vecinos más cercanos. De esta forma, se consideraría un método global. La desventaja es que el algoritmo se ejecutaría más lentamente.

2.6.3.3 Ventajas

- Robusto al ruido de los datos del entrenamiento [Fernandez, 2007] [Teknomo, 2006], especialmente si se tiene en cuenta el factor peso (w_{ij}).
- Efectiva si el número de datos de entrenamiento es grande.

2.6.3.4 Desventajas

- Es necesario determinar el valor del parámetro k [Fernandez, 2007] [Teknomo, 2006], es decir, el número de vecinos más cercanos.
- El aprendizaje basado en distancias no deja claro que tipo de distancia hay que usar y que atributos tienen que intervenir para obtener los mejores resultados.
- El coste computacional es elevado porque se necesita calcular la distancia de cada consulta con los ejemplos de entrenamiento. Algún algoritmo de indexación, K-Dtree, reduce el coste computacional.

2.6.3.5 Técnicas de reducción del conjunto de entrenamiento a aplicar en algoritmos basados en aprendizaje de ejemplos

Una de las características comunes entre los algoritmos basados en aprendizaje de ejemplos (k-NN, Teoría Bayesiana) [Ting Liu, 1999] [Fernandez, 2007] es que cada nuevo vector de entrada, necesita compararse con cada uno de las instancias que tenga el conjunto de datos. Esto deriva en una serie de problemas como que se requiere una gran memoria para el almacenaje de los ejemplos de training, lentitud en la clasificación y como consecuencia un gran coste computacional. Otro fallo a destacar, es que todas las muestras tomadas pasan a formar parte de la base de datos de training. Sin tener en cuenta, que algunas instancias pueden introducir ruido y degradar la capacidad de generalización.

Por estos motivos, es necesario introducir un proceso denominado técnica de reducción, que permita atenuar los efectos anteriores.

1. Teoría de la técnica de reducción

Los puntos a tener en cuenta para evaluar las diferentes estrategias para reducir un conjunto de datos son:

- a. Representación de vectores: Consiste en mantener los mismos valores del conjunto de entrenamiento o modificarlos usando una nueva representación, como normalización de datos y/o filtrado de ruido.
- b. Dirección de la búsqueda: La dirección influye cuando la técnica de reducción consiste en realizar un subconjunto a partir del conjunto de datos original, ya que, en función de donde esté colocada la instancia en el grupo de training, la probabilidad de que pase a formar parte del subconjunto varía. Se pueden diferenciar tres direcciones de búsqueda, incremental, decremental y por lotes.

c. Elección de la función de la distancia: Permite elegir los k vecinos más cercanos. Se ve influenciada por el peso de los atributos, así que hay que tener en cuenta los rasgos de éstos para intentar atenuar aquellos que sean más predominantes en la decisión. La normalización de los datos del vector es una forma de que el peso no sea determinante en la decisión.

d. Pesos: Algunos algoritmos usan esquemas de pesos que alteran las medidas de las distancias, votando la influencia de cada instancia.

2. Estrategias de evaluación

Hay una serie de propiedades que caracterizan a los algoritmos de reducción y a la efectividad de cada uno. Los criterios a los que se hace referencia son:

- Velocidad en la ejecución: Reducir el número de instancias almacenadas, supone decrementar el tiempo en buscar y clasificar un nuevo vector.
- Reducción del almacenaje: Uno de los principales objetivos consiste en reducir el número de atributos de cada instancia que se almacenan en el conjunto de datos.
- Exactitud de la generalización: Un algoritmo exitoso será aquel que permita reducir el número de vectores Fingerprint en el grupo de datos, sin perder capacidad de generalización.
- Tolerancia a ruido: Los algoritmos también difieren con respecto a lo bien que trabajan en presencia del ruido. Ante la presencia del ruido, hay dos problemas que pueden ocurrir. El primero es que muy pocas instancias sean borradas del set de entrenamiento porque son necesarias para mantener las barreras del ruido. El segundo problema es que la exactitud de generalización puede sufrir, especialmente si las instancias ruidosas se mantienen mientras que las instancias buenas se borran. En cada caso la reducción del conjunto de entrenamiento, puede ser menos exacto que el de entrenamiento completo al clasificar el vector de entrada.
- Tiempo de requerimientos: Para algunas aplicaciones reales la fase de entrenamiento si es muy larga, requiere la aplicación de una técnica de reducción, que no tiene una relación directa con el tamaño del grupo de instancias. Ya que en aquellos, en los que el conjunto de training es muy grande, el tiempo global de ejecución del sistema de localización es mayor con una técnica de reducción de instancias, que sin su presencia.
- Incrementación: Ante cambios en el entorno, sería idóneo, permitir que el grupo de training aceptara nuevas instancias. Cada algoritmo de reducción se focalizaría más en perfeccionar unos objetivos en detrimento de otros.

3. Algoritmos de reducción de instancias.

A continuación se explican técnicas para reducir el conjunto de entrenamiento [D. Randall Wilson, 2000] donde S se denomina como el nuevo conjunto a crear con las diferentes técnicas a partir de T que es el grupo de instancias de entrenamiento original.

Se podrá observar que se hace mención en repetidas ocasiones al algoritmo k -NN, ya que es el algoritmo con el que se trabaja en el caso de estudio.

- IB2 (Instance Based learning algorithm 2)

Es un método incremental. Comienza con S vacío y cada instancia en T se añade a S si no es clasificada correctamente por las instancias ya existentes en S (la primera instancia siempre se añade). Este algoritmo guarda los puntos límites en S mientras elimina los puntos internos que están rodeados de puntos que pertenecen a la misma clase. Es extremadamente sensible al ruido, porque las instancias erróneas (ruidosas) siempre serán clasificadas, mientras que los vectores fiables se borrarán.

- MCS (Model Class Selection)

El proceso para reducir el tamaño del conjunto consiste en llevar la cuenta de cuantas veces cada instancia fue uno de los k vecinos más cercanos de otra instancia, y si su clase es la misma que la de la instancia que está siendo clasificada. Si el número de veces que fue erróneo es más grande que el número de veces que fue correcto, se elimina el vector. Este método permite eliminar el ruido.

- CNN (Condensed Nearest Neighbour Rule)

El algoritmo consiste en formar un conjunto de instancias S a partir de T , donde cada miembro de S sea más cercano a un miembro de S de la misma clase que a uno de S de clase diferente.

1° Se eligen aleatoriamente una instancia de T de cada clase de salida y se añade a S .

2° Se clasifica cada instancia de T , tomando como vectores de entrenamiento el subgrupo S y con $k=1$. Si se clasifica correctamente, esta instancia que actúa como nuevo usuario, forma parte de S .

3° Este proceso se repite hasta que no quedan más instancias en T .

El algoritmo asegura, que todas las instancias de T se clasifiquen correctamente, aunque no garantiza un mínimo set.

El algoritmo CNN es muy sensible al ruido, ya que se tiende a clasificar en S las instancias ruidosas de T . La presencia del ruido causa que no se reduzca el conjunto de entrenamiento y que la capacidad de generalización disminuya, ya que si instancias de ruido pasan a formar parte de S , éstas son muestras outlier, es decir, casos puntuales que no se ajustan al comportamiento normal del resto de muestras de una clase. Aquí la presencia del ruido sería muy fuerte, ya que, clasificaría erróneamente.

- SNN (Selective Nearest Neighbour)

SNN dice que cada miembro de T debe ser más cercano a un miembro de S de la misma clase que a una instancia del subconjunto de una clase diferente. Este algoritmo también es sensible al ruido, aunque tiende a sacrificar el almacenamiento más que la exactitud cuando el ruido está presente.

- RNN (Reduced Nearest Neighbour Rule)

Gates presentó la reducción de la Regla para el algoritmo de vecinos más cercanos, RNN. El algoritmo comienza con $S = T$ y elimina cada instancia de S si esta eliminación no causa que otros casos en T sean clasificados incorrectamente por el resto de casos en S .

- ENN (Edited Nearest Neighbour Rule)

S es igual a T , pero S elimina aquella instancia que no esté de acuerdo con la clasificación de sus k vecinos más cercanos, normalmente $k=3$. Permite borrar las instancias ruidosas y así mismo reducir el número de vectores Fingerprint en el conjunto de instancias.

- All-kNN

Este algoritmo trabaja como se indica a continuación:

Para $i=1$ hasta k , se indica como incorrecta cualquier instancia no clasificada correctamente por sus " i " vecinos más cercanos. Después de completar el bucle k veces, borra cualquier instancia de S marcada como mal. All-kNN es el método más exacto entre ENN y RNN. Estos métodos se utilizan más como filtros de ruido que como de reducción.

- VSM (Variable Similarity Metric)

Este algoritmo borra aquellas instancias que tenga sus k vecinos de la misma clase, independientemente de si pertenecen al grupo del vector o a otro. VSM mantiene aquellos vectores que actúan como puntos limítrofes entre zonas adyacentes.

Se puede destacar, que utiliza valores de k grandes, por ejemplo $k=10$.

3.- DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Una vez que se han presentado las diferentes tecnologías y métricas para definir un sistema de localización, se van a elegir aquellos métodos que conforman el caso que se detalla a continuación.

3.1 ELECCIÓN DEL DISEÑO DEL SISTEMA

El escenario de desarrollo del proyecto es en la primera planta del edificio Torres Quevedo en el campus de Leganés de la Universidad Politécnica Carlos III de Madrid. Al tratarse dentro de una construcción, se trata en un entorno indoor y como consecuencia el sistema de localización se clasificará como tal.

A continuación se detalla los módulos en los que se divide el sistema de localización diseñado para adaptarse a las especificaciones del proyecto que se está tratando.

3.1.1 Bloques del sistema de localización

3.1.1.1 Estimación de los parámetros de error

Según el diagrama de bloques en el que se divide el sistema de localización que presentado en el apartado 2.1, el primer paso consiste en elegir la naturaleza de las señales recibidas, que en el caso que aquí se presenta son señales radio, en concreto WLAN estándar 802.11b. La información que se aprovecha es la potencia recibida de las señales, RSSI, por los AP's estáticos que se encuentran distribuidos por los diferentes pasillos de la planta del edificio.

Entre las métricas que se han estudiado en el apartado 2.3, RSSI es la más sencilla de utilizar ya que la capa MAC directamente proporciona información de la potencia emitida por los AP's cada cierto período de tiempo. Gracias a la técnica utilizada, WLAN, para desarrollar el sistema de localización, no es necesario implementar ningún hardware adicional, ya que la estructura existente junto a un software, completan la función. Esta ventaja, junto con que es una tecnología de las más extendidas entre numerosos edificios, son las razones de la elección de 802.11b para el desarrollo de este proyecto.

3.1.1.2 Estimación de la localización de dispositivos

Una vez que se han obtenido las señales RSSI, el siguiente paso, es determinar cómo se realiza la estimación de la ubicación y la representación de la misma. En este punto se focalizan los intereses del proyecto.

Se utiliza un algoritmo que permita la localización más óptima. En este caso, se detecta la posición por proximidad a un AP, tomando un sistema de referencia en dos dimensiones.

También se puede destacar que es un sistema de localización global, determinista y sin necesidad de sincronismo, ya que estima una ubicación en un tiempo t sin necesidad de conocer la posición en $t-1$. Esta posición puede ser física, determinada por dos coordenadas del eje cartesiano (x, y) o simbólica, es decir, identificada con una zona de cobertura definida.

La técnica seleccionada para este proyecto es Fingerprint, junto con el algoritmo k-NN en la fase Real-time, ya que como se explicó en el apartado 2.5.2, constituyen una buena alternativa para representar y procesar las medidas RSSI para estimar la localización.

3.1.2 Fases de localización

En función de las tareas que se deben de llevar a cabo por el sistema de localización, el proceso se puede dividir en tres fases, que son:

- Toma de muestras
- Fase Off-Line o de Training
- Fase Real-Time



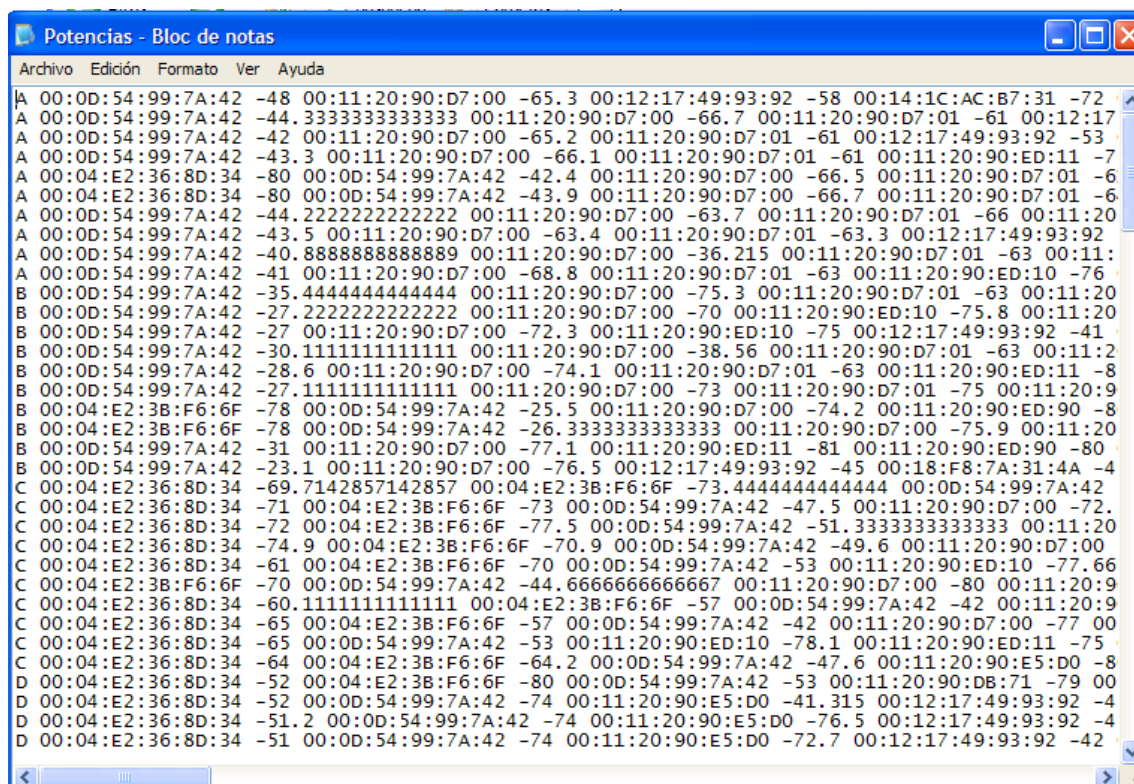


Figura 15: Volcado de los datos de training.

Se escoge la primera línea para explicar los componentes que lo forman:

A 00:0D:54:99:7A:42 -48 00:11:20:90:D7:00 -65.3 00:12:17:49:93:92 -58 00:14:1C:AC:B7:31 -72

A: Corresponde con la zona de cobertura donde se ha registrado la muestra.

00:0D:54:99:7A:42: Dirección MAC de un AP que ha emitido la potencia con el valor que le acompaña.

-48: Potencia recibida expresada en dB.

Se han recogido un total de 181 muestras. Estas muestras se van a dividir en dos grupos diferenciados por su función: training y testing.

3.1.2.2 Fase Off-Line o de Training

La sencillez de esta fase, es uno de los motivos por los que se eligió este método para implementar el núcleo del sistema de localización. Como se explicó en el apartado de algoritmos Fingerprint y k-NN, consiste en almacenar un conjunto de valores de los parámetros que se toman como referencia en un formato determinado. Este conjunto de valores se identifica con el grupo de training en el que se dividen las muestras tomadas.

Cada zona de cobertura está identificada por unos determinados AP's, de esta forma, no en todas las áreas se recogen potencias de los mismos equipos transmisores. Teniendo en cuenta esto, se puede definir la función Fingerprint, que elabora vectores con las potencias de cada muestra, de tal forma que identifiquen la zona donde se recibieron. El vector tiene una dimensión de uno por el número total de AP's que se haya registrado. Dibuja un plano bidimensional, porque sólo se hace referencia a una planta del edificio del Torres Quevedo.

Habrán tantos vectores como muestras conformen el grupo de training y un vector principal formado con todos los AP's donde se establece una relación entre la posición del vector y la dirección MAC correspondiente. Cada vector tendrá las potencias de cada muestra pero colocadas en la misma posición donde se encuentra la dirección MAC del AP que lo emitió en el vector principal. De esta forma se ha

conseguido un conjunto de datos, que permite junto a la segunda fase del algoritmo k-NN, localizar los posibles nuevos dispositivos móviles.

3.1.2.3 Fase Real-Time

En esta fase se aplica el algoritmo de localización k-NN, que a través de una medida de similitud o de diferenciación permite comparar los datos de un nuevo usuario a localizar con los vectores del conjunto de training. Los nuevos usuarios se identifican con el grupo de testing.

El proceso k-NN se divide a su vez en otras dos etapas, Training, que coincide con la Fase Off Time de la técnica Fingerprint y Ejecución, que comprende la fase de comparación en Real-Time y localización. Así, una vez que se ha construido el vector Fingerprint a localizar, la etapa en Real-Time se divide en los siguientes pasos característicos del algoritmo k-NN.

- Clasificación.

Se calcula la distancia que hay entre la instancia que ha reportado el dispositivo nuevo y las instancias del conjunto de training. Las k distancias más pequeñas serán los k vectores más cercanos al dispositivo a localizar.

- Localización

El siguiente paso consiste en determinar la ubicación de los vectores de testing a partir de los k Fingerprint seleccionados en el apartado anterior. Cada vector está asociado a su localización, que en función de su representación, el método de decisión variará. Si las áreas de cobertura se representan cualitativamente, la localización del nuevo usuario será el valor cualitativo que sea más presente entre los k resultados (que más se repita).

Si por el contrario, las áreas de cobertura se representan cuantitativamente, la ubicación será el resultado de una media aritmética entre las coordenadas de todos los k vecinos.

A las fases que se acaban de describir, se pueden añadir nuevos procesos cuyos propósitos sean depurar datos, optimizar resultados y disminuir la carga computacional. En este proyecto se han aplicado modificaciones de este tipo para alcanzar dichos objetivos.

3.1.3 Técnicas de optimización del sistema de localización.

Los métodos basados en ejemplos como es k-NN, en condiciones de un gran volumen de datos, tienen que tomar decisiones de reducción para disminuir la complejidad, tiempo computacional y atenuación de ruido.

En el caso de k-NN, se conservan todas las instancias de training, con la ventaja que el aprendizaje es muy rápido porque sólo necesita almacenar cada vector en una base de datos. Consigue gran exactitud, sin ningún proceso añadido al de training. Sin embargo, almacenar todas las instancias, requiere gran capacidad en la base de datos y lentitud durante el proceso de clasificación.

Otra consecuencia de almacenar todos los vectores Fingerprint, es que también se guardan las instancias ruidosas, que pueden degradar la exactitud del sistema de localización.

Las estrategias que se han tomado para paliar los efectos que degradan la efectividad del sistema de localización se dividen en dos grupos en función de la fase en la que se apliquen, según se describe a continuación:

3.1.3.1 Técnicas de optimización en la fase Real –Time

- Normalización

Cada una de las zonas en las que se ha dividido el mapa, está determinada por las potencias que se han recogido y en la posición que se toman. Un nuevo usuario a localizar medirá la diferencia entre sus potencias y las del grupo de entrenamiento a través de una medida de

similitud. La diferencia entre las potencias de una misma muestra que tenga una variación considerable con respecto al resto de diferencias, tomará un valor muy significativo en la clasificación de la zona, lo que lleva en algunos casos a equívocos. En el caso que se estudia y tomando únicamente valores numéricos, se decide aplicar la siguiente fórmula:

$$Potencia\ Normalizada\ (i)(j) = \frac{Potencia\ (i)(j)}{\sum Potencia\ (i)(j)}$$

Donde $i = 1$ hasta j y j = número de máquinas que se han identificado en el entorno.

Unido a esto, se puede aplicar otro tipo de normalización, donde no sólo se toma como único atributo las potencias de las señales, sino también, las posiciones de los mismos. Con este hecho, se intenta atenuar la importancia del lugar que ocupe la potencia. Así, se tienen dos tipos de atributos: numérico (potencia) y nominal (dirección MAC del AP que determina la posición).

- Distancias

La función distancia [Montero] se emplea como medida de similitud o de diferenciación, como método para decidir cuáles son los vecinos más cercanos a un nuevo Fingerprint de entrada que corresponde con los k vectores del grupo de entrenamiento que dan distancias menores. El algoritmo kNN, trabaja con diferentes distancias, ya que cada medida de similitud se caracteriza por unas propiedades que pueden favorecer o no a conseguir resultados óptimos. Las funciones distancias que se aplican en el caso de estudio son:

- Distancia Euclídea:

La distancia Euclídea, es la distancia más usada para el algoritmo k-NN. Se define como:

$$d_{ij} = \sqrt{\sum (x_i - x_j)^2}$$

donde $i=1$ hasta N , N = número de AP's que se han identificado en un entorno.

Adaptado al problema de localización:

$$d_{ij} = \sqrt{\sum (AP_{iConjuntoEntrenamiento} - AP_{iNuevo})^2}$$

donde $AP_{iConjuntoEntrenamiento}$ y AP_{iNuevo} son las potencias del Fingerprint del conjunto de entrenamiento y del nuevo vector de entrada, respectivamente.

La distancia Euclídea examina la diferencia de cuadrados entre las coordenadas, en este caso potencias, de un par de vectores. Esta función es apropiada cuando los atributos de entrada son numéricos y tienen rasgos de igual peso aproximadamente. Cuando los atributos tienen diferentes rasgos, los atributos pueden ser normalizados por distancias individuales, por rangos o desviación estándar del atributo.

Las funciones distancias que introducen estas variaciones y se estudian son las siguientes:

- Distancia City Block o Manhattan

Examina las diferencias absolutas entre dos coordenadas o dos atributos de dos objetos.

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}|$$

La distancia City Block es la distancia Minkowski* con $\lambda = 1$.

- Distancia Chevychef

También se denomina valor máximo de la distancia.

Examina la magnitud absoluta de la diferencia entre las coordenadas o atributos de un par de objetos.

$$d_{ij} = \max_k |x_{ik} - x_{jk}|$$

La distancia Chevychef es la distancia Minkowski* con $\lambda = \infty$ (tomando un límite).

- Distancia Canberra

Se basa en la suma de series de fracciones diferenciales entre dos puntos o atributos de dos objetos. Cada término de la fracción diferencial tiene valores entre 0 y 1. Si una de las coordenadas es 0, el término se convierte a unidad a pesar del otro valor, es decir la distancia no se verá afectada. Notar que si ambas distancias son 0, se necesita definir que $0/0 = 0$. La distancia Canberra es muy sensible a pequeñas variaciones cuando ambas coordenadas se encuentran cerca del valor cero.

$$d_{ij} = \sum_{k=1}^n \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}$$

- Distancia Bray Curtis

También se denomina distancia Sorensen como método de normalización.

Ve el espacio como una rejilla, similar a la distancia City Block. Bray Curtis tiene una buena propiedad, si todas las coordenadas o atributos de los objetos son positivas, su valor está entre 0 y 1. El cero de Bray Curtis significa que las coordenadas son exactamente iguales y la distancia es indefinida. La normalización se hace dividiendo la diferencia absoluta entre el sumatorio.

$$d_{ij} = \frac{\sum_{k=1}^n |x_{ik} - x_{jk}|}{\sum_{k=1}^n (x_{ik} + x_{jk})} \quad d_{ij} = \frac{\sum_{k=1}^n |x_{ik} - x_{jk}|}{\sum_{k=1}^n x_{ik} + \sum_{k=1}^n x_{jk}}$$

* La distancia Minkowski se define como:

$$d_{ij} = \lambda \sqrt[\lambda]{(x_{ij} - y_{ij})^2}$$

En función del valor que tome λ se puede diferenciar:

Si $\lambda=1$, se tiene la distancia City Block.

Si $\lambda=2$, se tiene la distancia Euclídea.

Si $\lambda=\infty$, se tiene la distancia Chevyshev

- Pesos

Para evitar la situación que un usuario no se pudiera localizar [Kim, 1999], porque no hubiera mayoría de zonas entre los k vecinos más cercanos, se introdujo el concepto de peso. Hasta el momento, se ha considerado que todos los k vecinos, independientemente del orden de elección, tenían la misma importancia a la hora de determinar la localización. Sin embargo, se puede atribuir un cierto peso en la decisión a aquel vector Fingerprint que se encuentre más cerca del usuario a localizar, sin tener en cuenta efectos de multipath y otros efectos que degraden la calidad de la señal, ya que normalmente menor tiempo de llegada significa estar situado más cerca de unos determinados AP's que ubican una zona o

situación. Partiendo de este hecho, ante la situación indefinida de un usuario, a posteriori de la realización de la evaluación, se asigna como localización por defecto la correspondiente al vecino $k=1$.

- Valor de k

Un parámetro característico del algoritmo k -NN que influye en gran medida a la decisión de localización [Petru, 2006], es el número de vecinos usados para proporcionar una salida ante un nuevo vector de entrada. Para determinar k , se usa la validación leave-one-out-cross. Para cada varios valores de k , cada instancia se clasifica por sus vecinos más cercanos. El valor de k que obtenga la precisión más alta, se elige. Sin embargo, k , en algunas ocasiones, tendría que tener un valor dinámico, ya que previo a reducir el número de instancias, se tiende a elegir valores de k diferentes de una para intentar eludir la presencia del ruido, pero posteriormente a la creación de un subconjunto del principal y donde las instancias que deterioran la señal se han eliminado, el valor idóneo de k sería 1, de otra forma, k se mantendría constante.

- Valor de N

Otro parámetro importante en el funcionamiento del algoritmo es el número de potencias que se tienen en cuenta para realizar la medida de similitud entre los valores de una nueva entrada y los de una instancia del conjunto de datos, debido a que cada zona se identifica por las señales RSSI y la posición de las mismas dentro del vector Fingerprint. Si se toman valores pequeños, puede suceder que ciertas áreas no se puedan identificar, ya que sus potencias características se encuentran en posiciones avanzadas de la matriz o si se considera N un número grande, se incluyan potencias ruidosas que no deberían de estar presentes en un área que influyan negativamente en la decisión de localización.

3.1.3.2 Técnicas de optimización en la fase Off-Line:

Este parámetro influye cuando se intenta substraer un subgrupo del conjunto de instancias de training. En este proyecto se toma en cuenta esta propiedad porque se implementan dos algoritmos de reducción de instancias: k -dd [M.F. Cátedra, 2007] y ENN.

- k -dd

Con este proceso se intenta reducir el número de instancias que forman el conjunto de training que clasifican las nuevas entradas. Los pasos que intentan alcanzar este fin son:

1° Se compara cada nuevo Fingerprint con cada una de las instancias iniciales que forman el conjunto de entrenamiento, para averiguar con cuál de los vectores de training tiene el mismo número de potencias que pertenezcan a los mismos AP's o el más mayoritario.

2° Una vez que se elija el vector de training con más señales de potencia que pertenezcan a los mismos AP's, se averigua la zona a la que pertenece este Fingerprint.

3° A continuación se forma un subconjunto, cuyas instancias sólo serán las que pertenezcan a la misma clase que el vector elegido en el segundo punto junto a ésta y los Fingerprint de zonas adyacentes.

Este subconjunto final será el grupo de instancias sobre el que se aplicará el proceso de clasificación. En este caso, la dirección de búsqueda que se ha implementado es incremental. Esto quiere decir que la posición de las instancias de training en el grupo de muestras etiquetadas es importante, porque aquellas que estén situadas en las primeras posiciones, tendrán más probabilidad de pasar a formar parte del conjunto final de muestras.

- ENN

Los pasos a seguir para la implementación de esta técnica, se ha descrito en el punto 2.6.3.5 donde se explican más algoritmos de técnicas de reducción. El objetivo de este proceso, consiste en eliminar las instancias ruidosas que puedan interferir en la decisión de

localización. Indirectamente, si se elimina algún vector, también se disminuye el tamaño del conjunto de training.

3.2 IMPLEMENTACIÓN DEL SISTEMA

La técnica Fingerprint y el algoritmo de clasificación k-NN que forman el sistema de localización se han implementado en el lenguaje orientado a objetos Java. El sistema completo está formado por una serie de módulos cuya relación entre ellos se puede observar en el siguiente diagrama de flujo.

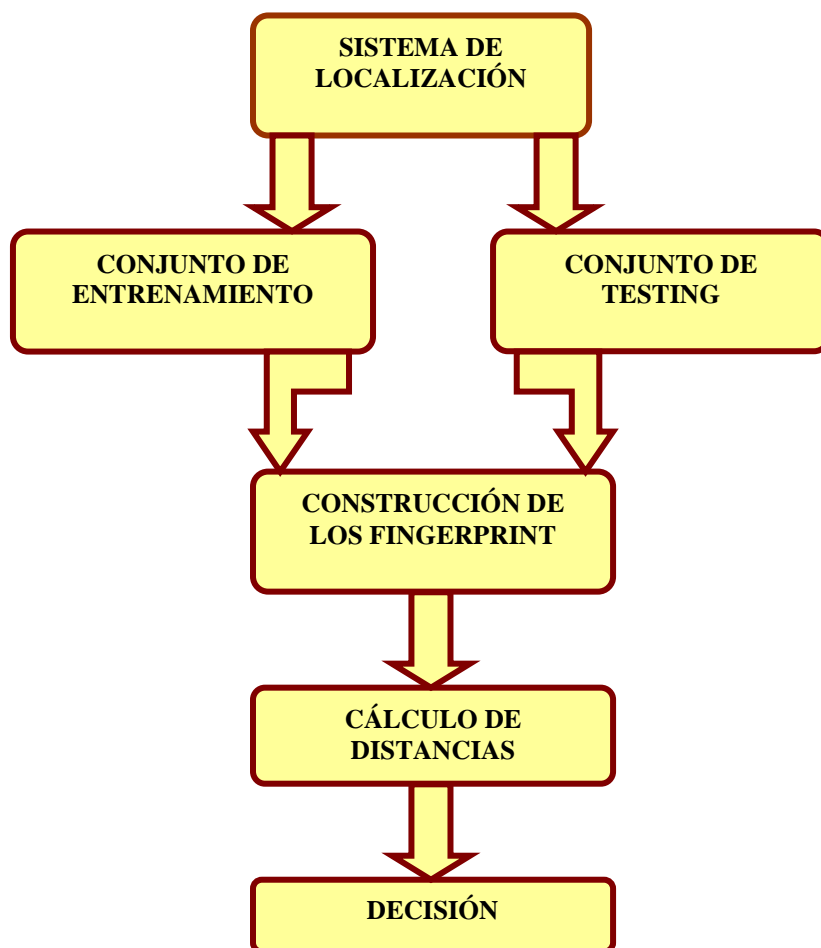


Figura 5: Diagrama de flujo.

A continuación se detalla cada uno de los módulos que forman el diagrama de flujo anterior. El diagrama de flujo que se ha explicado se identifica con la técnica Fingerprint y el algoritmo k-NN básico.

3.2.1 Conjunto de entrenamiento y de testing

La recogida de muestras que forman el conjunto de entrenamiento y de testing se ha realizado a partir de un script instalado en el equipo que recibe las señales de los puntos de acceso. Este script permite guardar las muestras en un fichero de texto, según el formato descrito en 3.1.2.1.

Para que esta información sea comprensible para el siguiente módulo, se necesita que todas las instancias guarden el mismo formato. Para ello, se realizan modificaciones tales como eliminar espacios, saltos de carro o separar potencias de la dirección MAC contigua.

3.2.2 Construcción de los Fingerprint

La función de este módulo consiste en ordenar la información que proporciona cada instancia de entrada, de tal forma que se obtenga un vector Fingerprint. Cada posición del vector Fingerprint está

asociada con un AP determinado que se ocupará por la potencia recibida de dicho AP. Estos vectores, son los términos que se utilizan para calcular la distancia de un nuevo elemento móvil a los puntos de acceso presentes en el entorno.

También se forma una matriz, denominada matriz de posiciones, donde se almacenan las localizaciones de las instancias del grupo de training. El número de la fila de la matriz de posiciones y del array donde se almacenan los vectores Fingerprint, guarda una relación bidireccional, es decir, la instancia que se guarda en la fila “i” del array de potencias RSSI, siempre encuentra su posición en la fila “i” de la matriz de posiciones y viceversa. Este procedimiento se necesita para la etapa de clasificación que se explica en el módulo de decisión.

El módulo de ordenación de datos, se asocia a la clase Java Fingerprint, donde se desarrollan los métodos necesarios para desempeñar la función que se ha descrito en el párrafo anterior.

3.2.3 Cálculo de distancias

En este momento del proceso, se calcula la distancia del vector Fingerprint del dispositivo a localizar con cada uno de los vectores Fingerprint del conjunto de entrenamiento. Para ello, se implementa la clase Distancia, junto con cinco tipos de distancias que se pueden utilizar para el algoritmo, que son la distancia Euclídea, City Block, Chevychev, Canberra y Bray Curtis.

Cada resultado se compara con los que se han obtenido previamente, para sólo almacenar tantas distancias más cercanas como el valor de k indica, es decir, los k vecinos más cercanos al nuevo dispositivo móvil.

Los métodos que actúan como interfaz entre el cálculo de distancias y el almacenaje de su resultado se encuentran en la clase KNNOrdenado.

3.2.4 Decisión

Se determina el área en el que se ubica el nuevo dispositivo móvil. Para ello, se utiliza la asociación que se realizó en el módulo de ordenación de datos, es decir, cada una de las instancias que ha proporcionado uno de los k vecinos más cercanos, se identifica con su ubicación.

Una vez determinadas las zonas en las que se encuentra cada vecino, se puede determinar la localización del dispositivo desconocido que será aquella área de cobertura más repetida entre los k vecinos. Si se llega a dar un situación en la que cada vecino pertenece a una localización diferente, se elige como ubicación del nuevo dispositivo la del vecino más cercano de entre los k que se han calculado, es decir, se realiza una decisión con la propiedad del peso.

3.2.5 Otras clases adicionales

Se han implementado otras funciones adicionales para optimizar los resultados del sistema de localización. Estas funciones son la normalización de potencias RSSI de cada instancia. La clase KNNOrdenado llama a la clase Normalización para realizar dicha modificación y obtener los nuevos datos que sirven como entrada del módulo procesado de datos.

La otra clase que implementa la técnica de reducción del conjunto de entrenamiento k-dd se llama Dinámico. La ejecución de sus métodos implementa los pasos que se explicaron en el punto 3.1.3.2, en el apartado k-dd.

4.- EVALUACIÓN DEL SISTEMA

En este capítulo se detalla la batería de pruebas con la que se va a validar el sistema de localización, que se ha presentado en el capítulo anterior. Las pruebas que se van a realizar consisten en modificar los valores que toman los parámetros del algoritmo y que influyen en el funcionamiento del sistema, que son:

- N: Número de medidas de potencias que se toman para realizar la distancia de similitud.
- k: Número de vecinos más cercanos.
- Distancias: Medidas de diferenciación para obtener los k vecinos más cercanos.
- Pesos: Propiedad que determina la importancia de cada una de las medidas tomadas. Se representa como la elección de vecino más cercano ($k=1$) en caso de ambigüedad.

También se normalizan los datos y se incluyen técnicas de reducción de muestras, como se ha explicado en el punto de técnicas de optimización en la fase Off-Line (ver 3.1.3.2), con el fin de optimizar el funcionamiento de la técnica Fingerprint y el algoritmo k-NN.

Las pruebas anteriores se van a realizar sobre tres experimentos distintos. Los experimentos se diferencian por el número de instancias que componen el grupo de entrenamiento y de evaluación. De esta forma se pueden distinguir tres grupos, 80%-20%, 70%-30% y 60% de muestras de training, 40% de testing respectivamente.

La evaluación del sistema se divide en tres fases que son:

- FASE I: Sistema de localización con 80% de instancias de entrenamiento.

La evaluación del sistema comienza con el conjunto de 20% de instancias de testing y la técnica Fingerprint y k-NN básico, es decir, no se realiza ningún tipo de filtrado de instancias y la distancia que se utiliza es la distancia Euclídea. Sobre estas bases se producen modificaciones en los valores de k y N. El siguiente paso consiste en estudiar los resultados que se obtienen al introducir la presencia del peso y de la normalización mientras se siguen variando los valores de los parámetros del número de vecinos y AP's. Y por último, se realizan los mismos experimentos, pero sustituyendo la distancia Euclídea por el resto de medidas de diferenciación que se han visto en el apartado 3.1.3.1.

- FASE II: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento.

Las pruebas que se han descrito en la primera fase, se aplican sobre los otros dos grupos de entrenamiento y testing, cuyos resultados se comparan con los que se han obtenido en el conjunto de training de 80% y 20% de instancias a localizar.

- FASE III: Optimización del algoritmo.

Finalmente la evaluación concluye cuando se introducen dos nuevos métodos con el fin de optimizar la técnica Fingerprint y k-NN básico. Sobre este sistema, también se aplican variaciones en los valores de k y N y se incluyen los atributos del peso y de la normalización para cada una de las diferentes distancias.

A continuación se detalla estos experimentos y los resultados obtenidos.

4.1 FASE I: Sistema de localización con 80% de instancias de entrenamiento

4.1.1 K y N variables

N: Variando este parámetro, se intenta obtener el grado de importancia del número de AP's que se toman en cuenta para obtener el resultado.

Según se muestra en la figura siguiente, con pocas potencias, las zonas no se identifican correctamente. Los resultados apenas superan el 50% de acierto y no se ven influidos por el número de k-vecinos elegidos, ya que los mejores aciertos se alternan entre $k=5$ y $k=7$.

Si se aumenta el número de potencias a medir a $N=36$, los resultados obtenidos mejoran notablemente.

Por último no se hace ninguna discriminación de potencias, es decir, $N=68$, donde el ratio de aciertos alcanza su máximo valor.

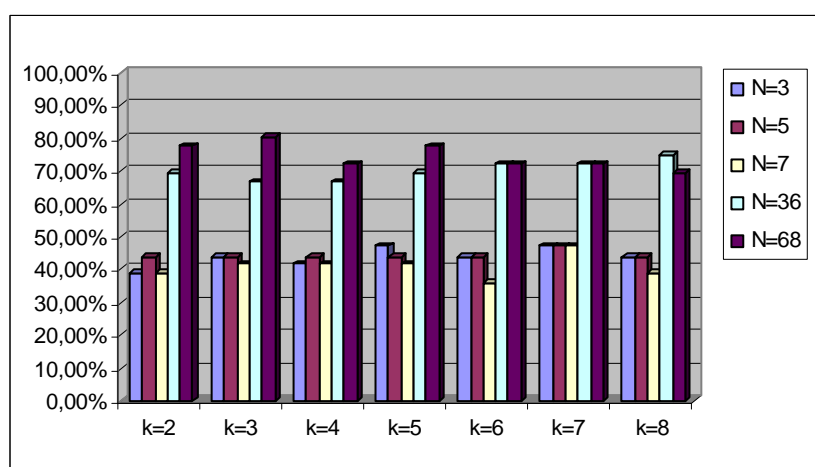


Figura 16: Estudio del comportamiento del algoritmo variando los parámetros k y N.

Una vez que se establece el número de potencias que determina cada punto de localización, otra decisión que se tiene que tomar, es el número de vecinos usados para determinar la localización ante un nuevo vector de entrada.

Siguiendo la técnica de validación leave-one-out-cross se han tomado valores desde $k=2$ hasta $k=8$, en el grupo de training de 80%, ya que el número máximo de muestras de una misma zona son ocho. Los mejores resultados que se han obtenido son con valores de k impares, ya que había menos probabilidad de que se diera una situación de indecisión porque no hubiera mayoría de zonas en los vecinos con menores distancias. Entre los k impares, se destaca como mejor resultado $k=3$, según se muestra en la siguiente figura.

A medida que se aumenta el número de vecinos más cercanos, la distancia mínima aumenta y así mismo la posibilidad de coger muestras que pertenezcan a áreas no correctas y contiguas, ya que la señal de potencia es más fuerte e interfiere en forma de ruido en la señal de la zona a localizar, es decir, se hace más sensible al ruido.

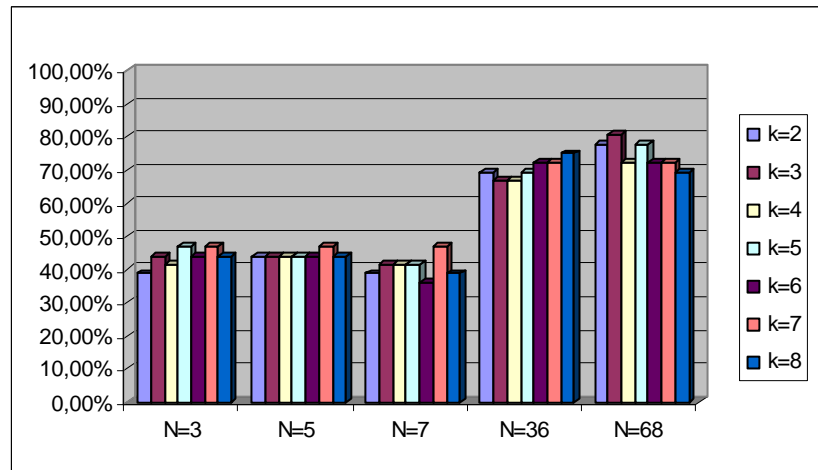


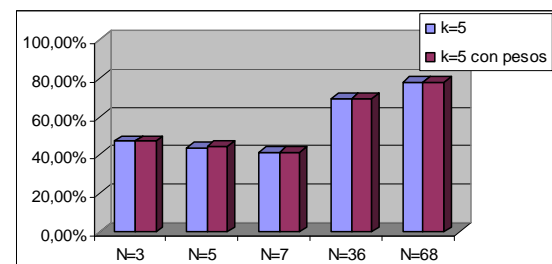
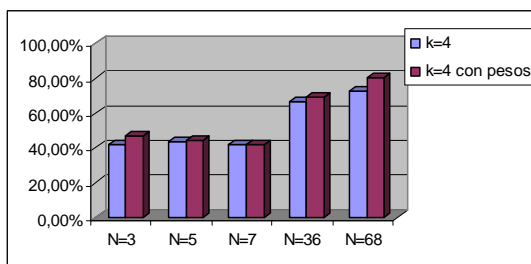
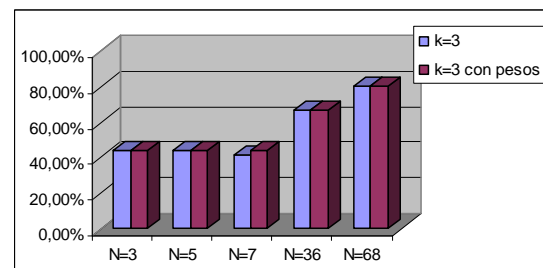
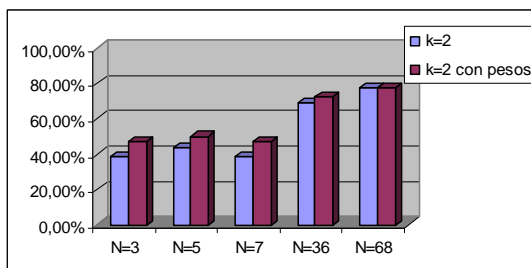
Figura 17: Estudio del comportamiento del algoritmo variando los parámetros k y N.

4.1.2 Pesos

Este experimento consiste en seleccionar la ubicación del vecino más cercano ($k=1$) cuando la salida de k-NN es una indeterminación.

Ante la presencia del peso el porcentaje de éxito en valores con k pares aumenta notablemente, al contrario que cuando k toma valores impares, excluyendo a $k=7$. El hecho de que $k=7$ no tenga el mismo comportamiento que el resto de k impares, se puede entender como que teniendo en cuenta 7 vecinos, se están cogiendo muestras frontera de zonas contiguas a la correcta, que provocan el efecto de la indeterminación en el resultado.

Esta variante, introduce también otros cambios, como que k, tomando valores pares, ofrece resultados mejores o iguales que los impares y en concreto para $k=4$. Este comportamiento se da, porque con k pares es más fácil que se dé una indeterminación en el resultado, porque no haya ninguna zona predominante. Al forzar a tomar un valor, ya se está obteniendo una localización y la mayor parte de las veces es acertada por tomar el vecino más cercano de entre los k.



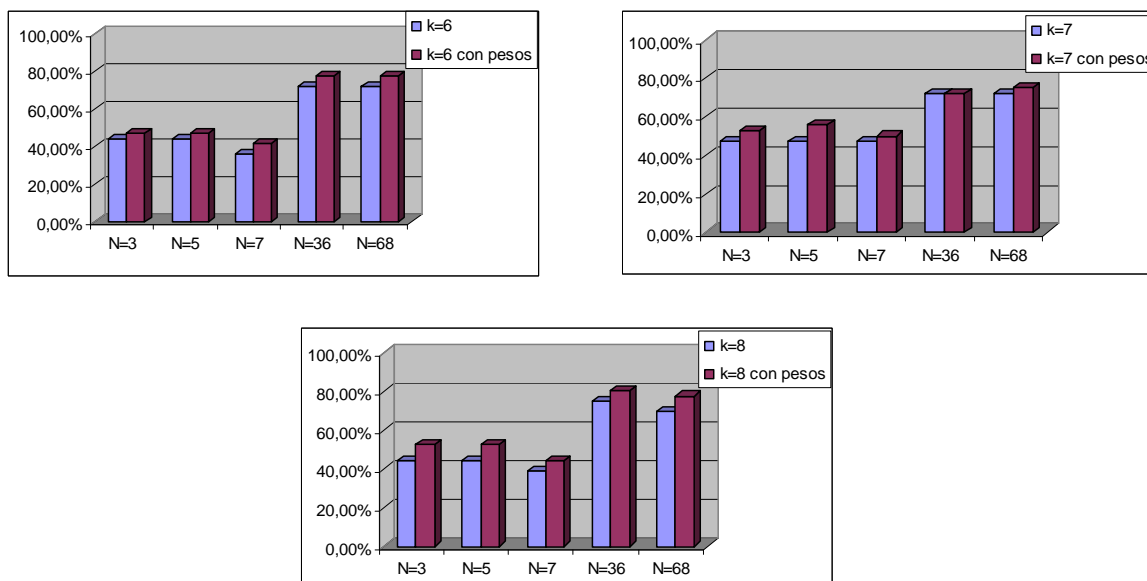


Figura 18: Comparativa del comportamiento del algoritmo sin la propiedad del peso y con ella para $k \in [2,8]$.

4.1.3 Distancias

Este experimento consiste en determinar la localización de un usuario, ejecutando el algoritmo k-NN con diferentes distancias.

Como se puede observar en la gráfica, la distancia que da un porcentaje más alto de localizaciones exitosas es la distancia Canberra. En estos resultados, no se ha tenido en cuenta el peso de cada uno de los vecinos más cercanos.

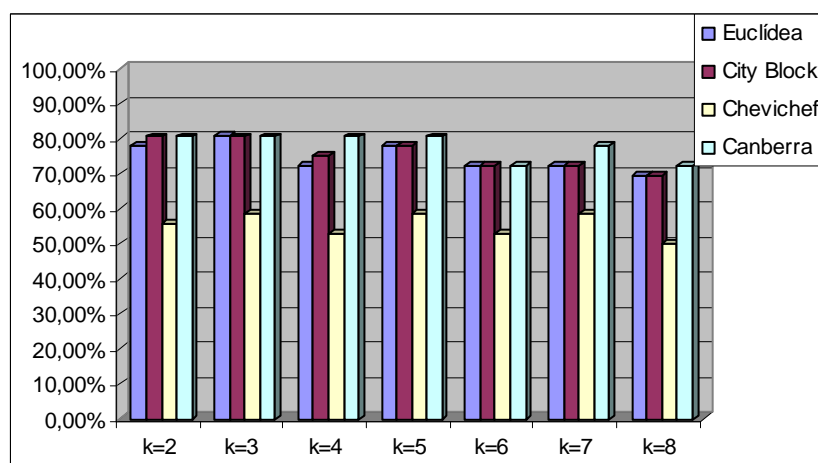


Figura 19: Evaluación de diferentes distancias sin $k=1$, sin normalizar y $N=68$.

En la gráfica que viene a continuación, sí se tiene presente la posibilidad de elegir la propiedad del peso ($k=1$). El resultado afecta a los k pares, ya que la distancia Canberra se ve superada por City Block y Euclídea, en este orden.

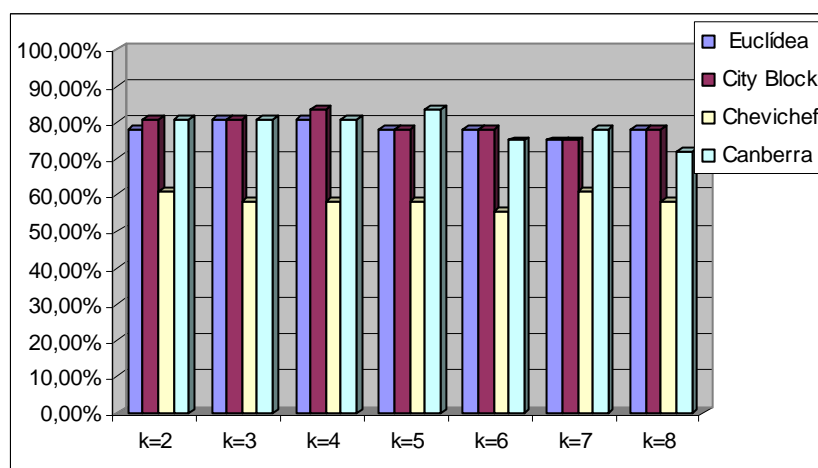


Figura 20: Comparativa entre diferentes distancias con la propiedad el peso y sin normalizar.

En general, Canberra, obtiene mejores resultados, porque según su fórmula al dividir por el factor $|x_{ik}| + |x_{jk}|$ permite que las diferencias entre coordenadas, en este caso, potencias, sean más suaves, evitando que un término sea más predominante que otro, sobre todo en presencia de ruido.

Por ejemplo, si tenemos $x_{ik} = -90$ y $x_{jk} = 0$. La distancia Euclídea sería:

$$\sqrt{(-90 - 0)^2} = -90$$

Se obtiene un resultado más predominante.

Mientras que con la distancia Canberra:

$$\frac{|-90 - 0|}{|-90| + |0|} = 1$$

Se obtiene un resultado menos predominante.

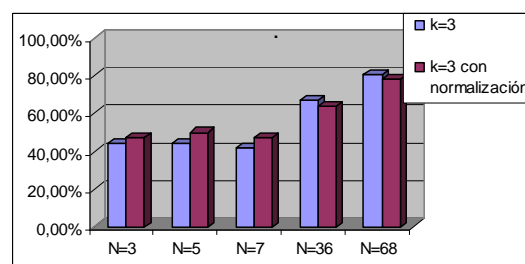
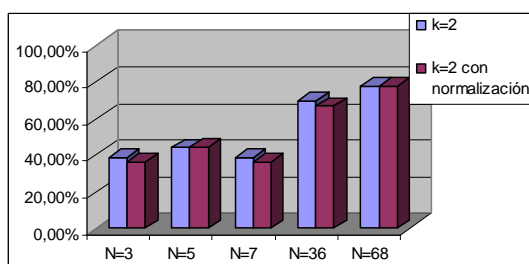
Se puede decir que la distancia Canberra, introduce el proceso de normalización.

4.1.4 Normalización

El proceso de normalización, se ha aplicado en los experimentos anteriores, es decir, en las variaciones de k, N, pesos y distancias. Los resultados se discuten a continuación.

4.1.4.1 Comparativa del comportamiento del algoritmo sin y con normalización de datos para k ∈ [2,8]

Excepto para k=2, la normalización produce mayor porcentaje de aciertos con pocas potencias (N=3, N=5 y N=7) que sin normalización. Al contrario que con N=36 y N=68, que predomina la barra de los datos no normalizados. Como se ha comentado previamente, con este proceso se intenta atenuar la importancia del resultado de la diferencia entre dos potencias y la posición de las mismas.



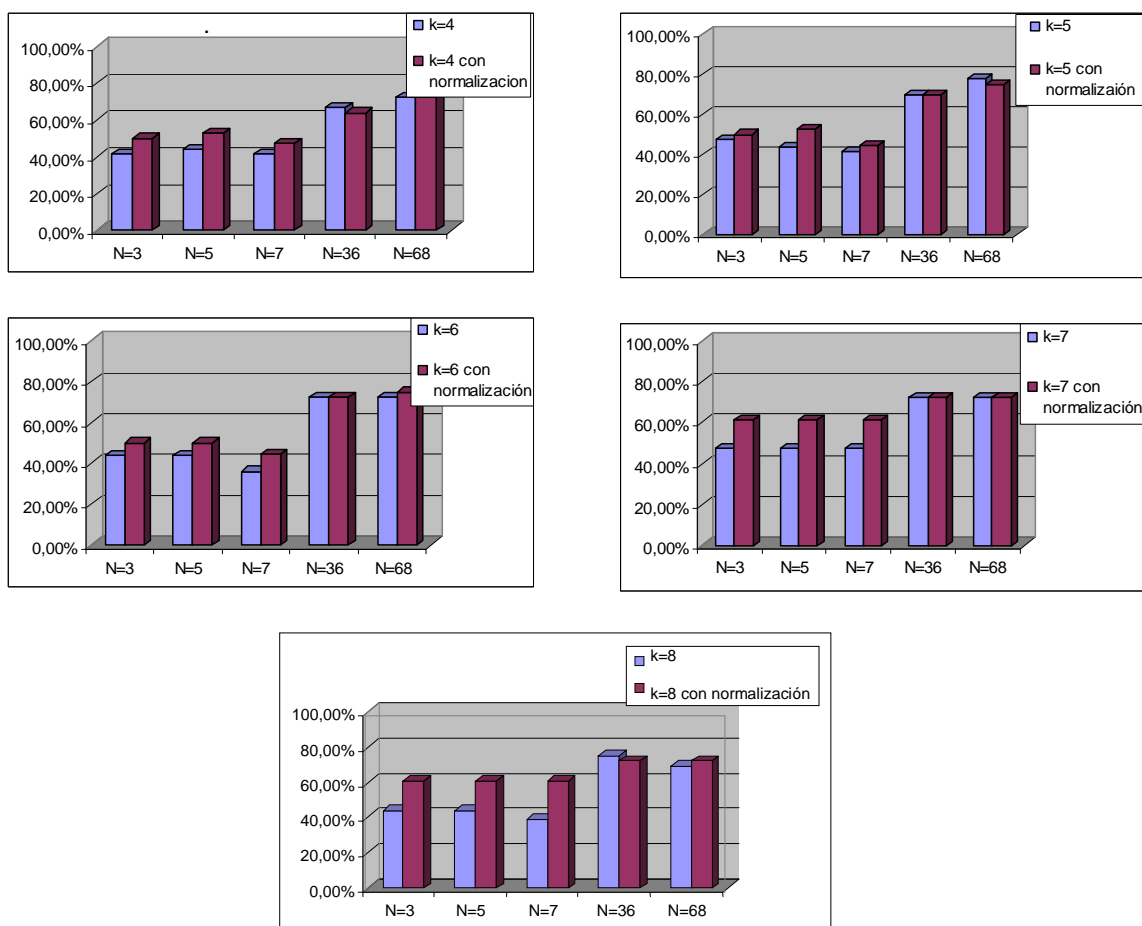


Figura 21: Comparativa del comportamiento del algoritmo sin y con normalización de datos para $k \in [2, 8]$.

Según los resultados, sólo se consigue quitar importancia a la posición de las potencias y a la diferencia entre dos de ellas, cuando se toman pocas potencias, porque esta elección tiene el mismo significado que la normalización, es decir, independiente de la llegada de las coordenadas. Sin embargo, cada zona se identifica por sus AP's y como consecuencia cuantas más potencias se tomen, más fácilmente se reconocerá una zona y así mismo la localización del usuario. Por este motivo, el caso de estudio es dependiente de las posiciones de las señales RSSI y si se intenta romper esta relación los resultados empeoran.

La diferencia de la normalización que se aplica en este apartado a los datos con la normalización que ofrece Canberra, es que ésta atenúa la diferencia de distancias, que en un entorno ruidoso, como es en el que se trabaja, optimiza el porcentaje de acierto, pero sigue siendo dependiente de los lugares que se ocupan. Al contrario que los otros tipos de normalizaciones que se aplican, que rompen la relación directa con la ubicación de las coordenadas. La incorrelación de resultados, se puede atribuir a que el proceso de normalización se aplica durante el cálculo de la distancia de Canberra entre dos potencias o a las muestras del grupo de training, provocando el efecto que se ha comentado en el párrafo anterior.

4.1.4.2 Comportamiento del algoritmo con la propiedad del peso y con los datos normalizados

Las pruebas realizadas con estas variantes, confirman que los mejores resultados con diferentes valores de k se da cuando el número de AP's, que identifica cada zona es completa.

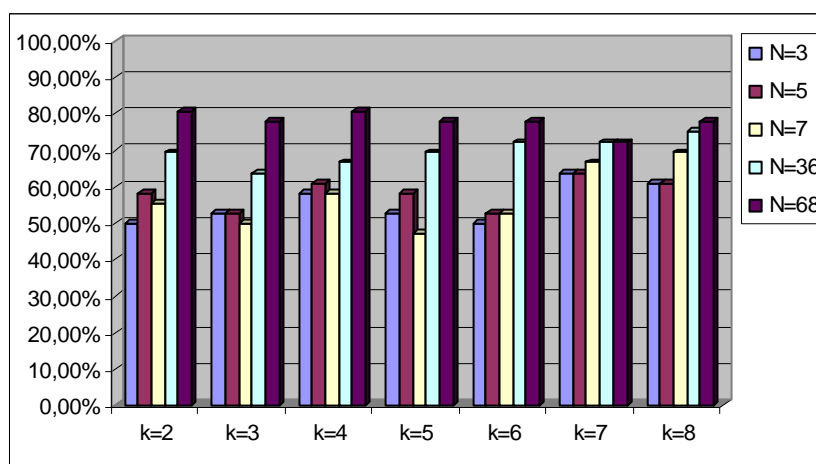
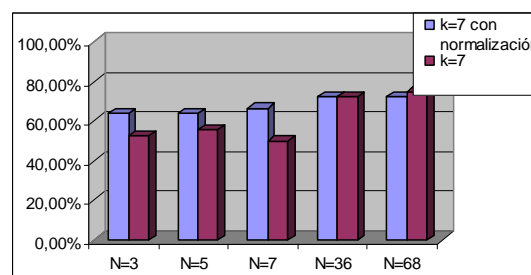
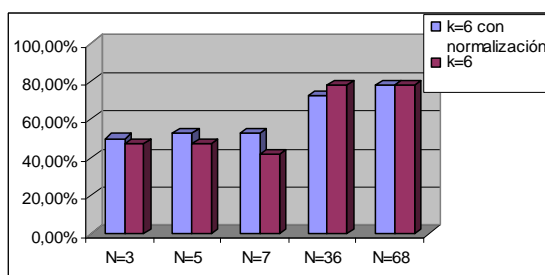
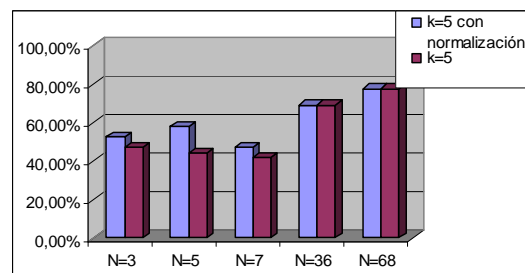
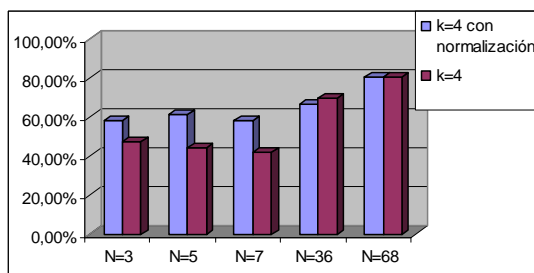
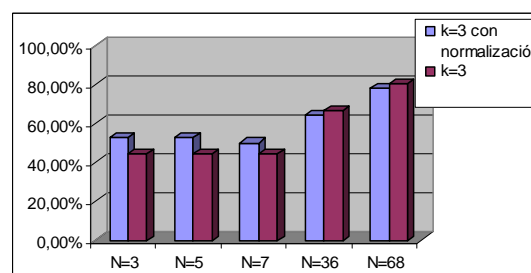
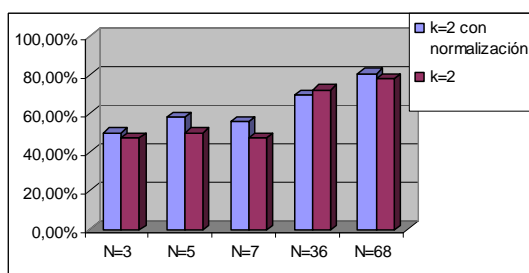


Figura 22: Comportamiento del algoritmo con la propiedad del peso y con los datos normalizados.

Para comprobar si se ha producido alguna mejora con respecto a añadir el proceso de normalización teniendo en cuenta el peso, se ha comparado con la misma variante pero sin normalizar. Las observaciones son muy semejantes al caso anterior es decir, ejecutando la normalización se obtienen más localizaciones exitosas con un número de potencias pequeños y se invierte o iguala el porcentaje de ubicaciones correctas con $N=36$ y $N=68$.

El razonamiento es el mismo que se ha explicado en el experimento que le precede.



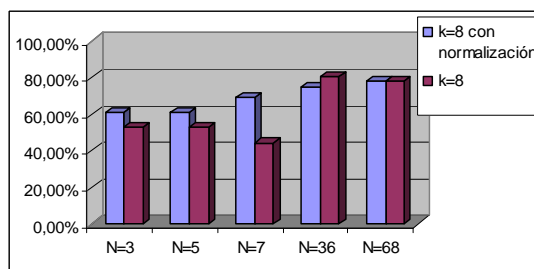


Figura 23: Comparación del comportamiento del algoritmo usando la propiedad del peso y con y sin normalización para $k \in [2, 8]$.

4.1.4.3 Comparativa de la importancia de la presencia del peso en el caso de los datos sin normalizar y normalizados

A continuación se intenta comprobar si el efecto al tener en cuenta el peso de la posición del AP es el mismo para los datos normalizados que sin normalizar.

Para el número de vecinos 2, 3, 4 y 5, la presencia el peso en los atributos es más importante en los datos normalizados que sin normalizar, al contrario que con $k=6, 7$ y 8. Se puede deducir que la normalización suaviza más la posición de aquellos valores que estén más alejados de una zona centro, ofreciendo la misma información que el resto de muestras y sobre todo sin sobrescribir las potencias principales de cada vector. De esta forma, las indeterminaciones en los resultados son menos numerosas, y como consecuencia, el porcentaje de acierto no mejora tan notablemente con la presencia del peso, como con valores de potencias sin haber variado su unidad, con un número de vecinos a tener en cuenta más reducido.

Tabla 2: Comparativa de efectividad de la presencia del peso en los datos normalizados y sin normalizar.

	N=3	N=5	N=7	N=36	N=68
k=2 sin peso	36,10%	44,40%	36,10%	66,70%	77,78%
k=2 con peso	50,00%	58,30%	55,60%	69,40%	80,56%
con normalización	-13,90%	-13,90%	-19,50%	-2,70%	-2,78%
k=2 sin peso	38,89%	44,00%	38,89%	69,40%	77,78%
k=2 con pesos	47,42%	50,00%	47,20%	72,20%	77,78%
sin normalizar	-8,53%	-6,00%	-8,31%	-2,80%	0,00%
	N=3	N=5	N=7	N=36	N=68
k=3 sin peso	47,20%	50%	47,20%	63,90%	77,78%
k=3 con peso	52,78%	52,78%	50,00%	63,89%	77,78%
con normalización	-5,58%	-2,78%	-2,80%	0,01%	0,00%
k=3 sin peso	44,00%	44,00%	41,67%	66,70%	80,60%
k=3 con pesos	44,00%	44,40%	44,00%	66,67%	80,56%
sin normalizar	0,00%	0,00%	-2,33%	0,03%	0,04%
	N=3	N=5	N=7	N=36	N=68
k=4 sin peso	50,00%	53%	47,20%	63,90%	72,20%
k=4 con peso	58,30%	61,10%	58,30%	66,67%	80,56%

con normalización	-8,30%	-8,30%	-11,10%	-2,77%	-8,36%
k=4	41,67%	44,00%	41,67%	66,70%	72,20%
k=4 con pesos	47,42%	44,40%	41,67%	69,40%	80,56%
sin normalizar	5,75%	0,40%	0,00%	2,70%	8,36%
	N=3	N=5	N=7	N=36	N=68
k=5 sin peso	50%	53%	44,40%	69,40%	75,00%
k=5 con peso	52,78%	58,30%	47,20%	69,40%	77,78%
con normalización	-2,78%	-5,60%	-2,80%	0,00%	-2,78%
k=5	47,20%	44,00%	41,67%	69,40%	77,78%
k=5 con pesos	47,42%	44,40%	41,67%	69,40%	77,78%
sin normalizar	0,22%	0,40%	0,00%	0,00%	0,00%
	N=3	N=5	N=7	N=36	N=68
k=6 sin peso	50%	50%	44,40%	72,20%	75,00%
k=6 con peso	50,00%	52,78%	52,78%	72,20%	77,78%
con normalización	0,00%	-2,78%	-8,38%	0,00%	-2,78%
k=6 sin peso	44,00%	44,00%	36,10%	72,20%	72,20%
k=6 con pesos	47,42%	47,20%	41,67%	77,78%	77,78%
sin normalizar	3,42%	3,20%	5,57%	5,57%	5,58%
	N=3	N=5	N=7	N=36	N=68
k=7 sin peso	61,10%	61,10%	61,10%	72,20%	72,20%
k=7 con peso	63,89%	63,89%	66,67%	72,20%	72,20%
con normalización	-2,79%	-2,79%	-5,57%	0,00%	0,00%
k=7 sin peso	47,20%	47,00%	47,20%	72,20%	72,20%
k=7 con pesos	52,78%	55,56%	50,00%	72,20%	75,00%
sin normalizar	5,58%	8,36%	2,80%	0,00%	2,80%
	N=3	N=5	N=7	N=36	N=68
k=8 sin peso	61,10%	61,10%	61,10%	72,20%	72,20%
k=8 con peso	61,10%	61,10%	69,40%	75,00%	77,78%
con normalización	0,00%	0,00%	-8,30%	-2,80%	-5,58%
k=8 sin peso	44,00%	44,00%	38,89%	75,00%	69,40%
k=8 con pesos	52,78%	52,78%	44,40%	80,56%	77,78%
sin normalizar	8,78%	8,78%	5,51%	5,56%	8,38%

4.1.4.4 Comparativa entre las diferentes distancias sin peso y con normalización

Como se puede observar en la gráfica, la distancia que ofrece mejores resultados, una vez que se han normalizado los datos, es la distancia City Block, excepto para $k=3$ que es Canberra.

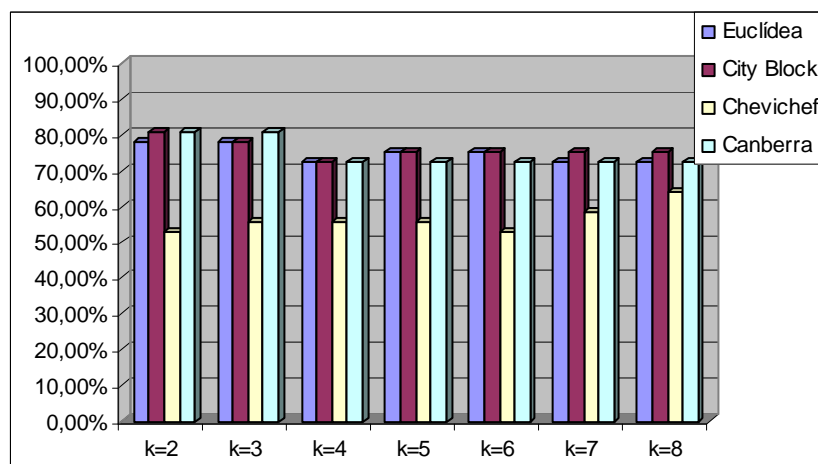
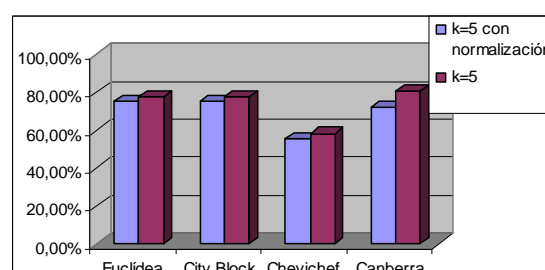
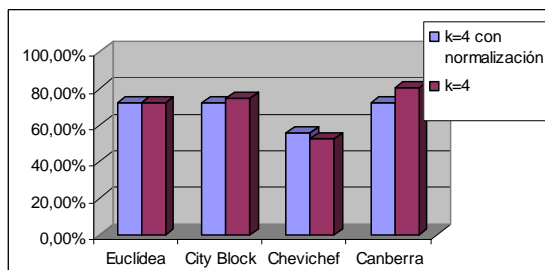
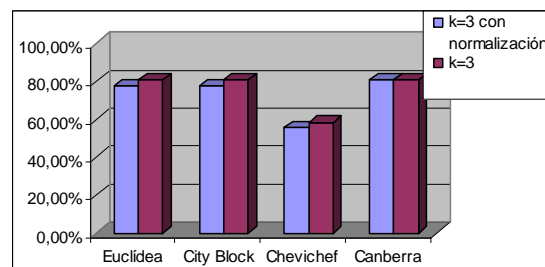
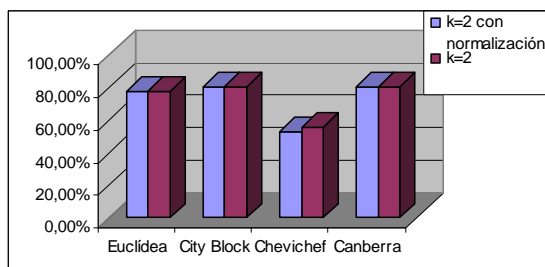


Figura 24: Comparativa entre las diferentes distancias sin peso y con normalización.

4.1.4.5 Comparativa entre las diferentes distancias sin la propiedad del peso, normalizando y sin normalizar los datos

Si estos resultados se comparan con las mismas medidas pero sin normalizar, se puede observar que los mejores porcentajes se atribuyen a la distancia Canberra para k igual a 2, 3, 4 y 5. Y sin embargo para un número de vecinos 6, 7 y 8, los resultados producto de la normalización, son más altos que sin introducir esta variación. Este ejemplo, confirma el razonamiento que se ha desarrollado en el punto anterior, donde la normalización es efectiva cuando los vecinos elegidos son frontera con otras zonas contiguas a la de interés para cada instancia.



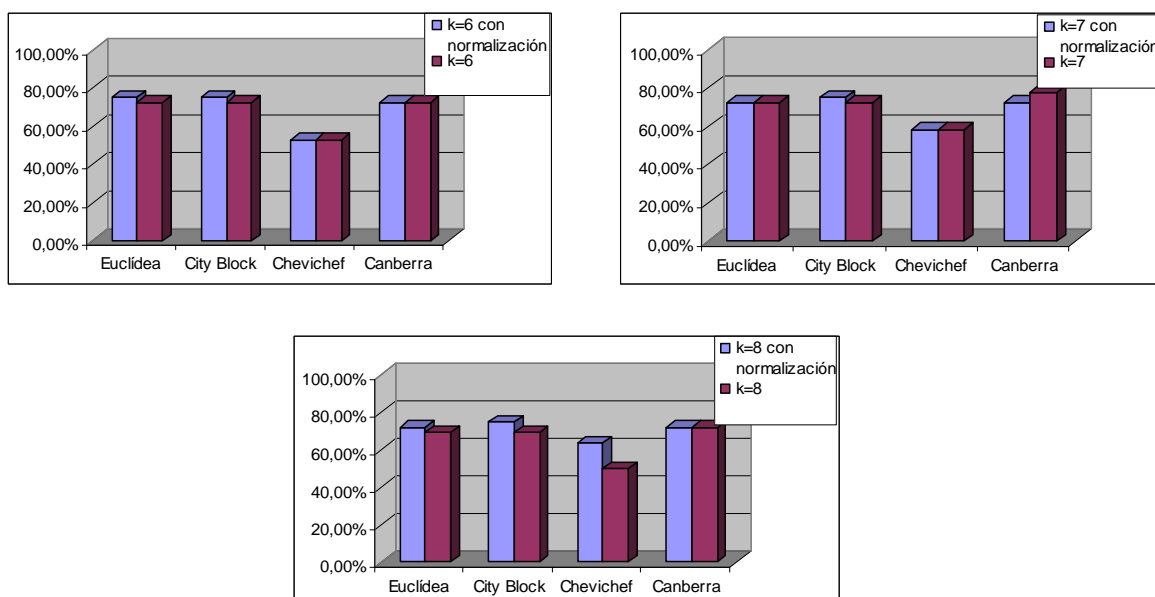


Figura 25: Comparativa entre las diferentes distancias sin la propiedad del peso, normalizando y sin normalizar para $k \in [2, 8]$.

4.1.4.6 Comparativa entre diferentes distancias con la propiedad del peso y normalizando

A continuación se comprueba el comportamiento del algoritmo ante la posibilidad de elegir como ubicación la del vecino más cercano ($k=1$), en el caso de una indeterminación.

La distancia que mejor acepta las variaciones sigue siendo City Block, excepto con $k=3$ que es Canberra, es decir, se mantiene el resultado anterior.

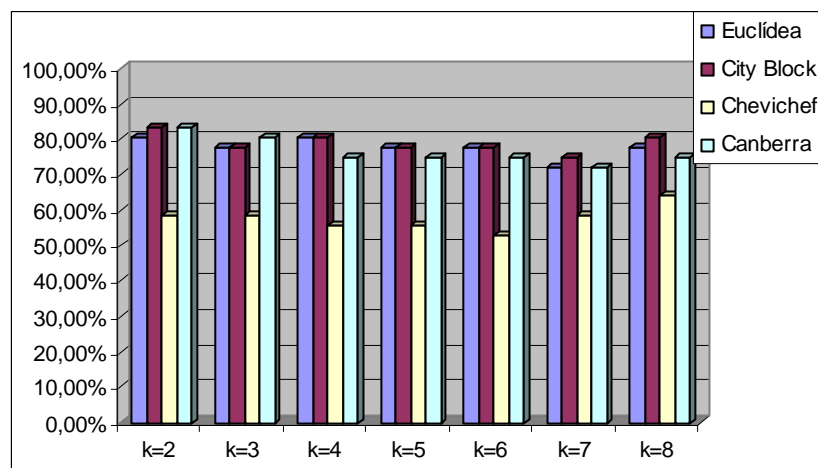


Figura 26: Comparativa entre las diferentes distancias con la propiedad del peso y normalizando.

4.1.4.7 Comparativa entre las diferentes distancias normalizadas con y sin peso para $k \in [2, 8]$

Ahora se comprueba la variación en los resultados con la presencia del peso y sin ella.

Se puede observar, que el porcentaje de acierto se ve beneficiado por la presencia del peso, para todos los valores de k y de las diferentes distancias, ya que o se incrementa o se mantiene.

Las distancias City Block, Chebichef y Canberra se comportan de modo similar que la distancia Euclidean ante la propiedad del peso, es decir, los porcentajes de éxito de localización que sufren un

incremento mayor son los que corresponden con k pares, a diferencia de los k impares, que en algunos casos como $k=7$, el incremento es nulo para todas las distancias.

Las distancias a las cuales les afecta más la presencia del peso son la distancia Euclídea y City Block, aunque a modo general, no son incrementos considerables. Se puede deducir, que una vez que los datos están normalizados, el número de indeterminaciones que se resuelven con la presencia el peso no es concluyente.

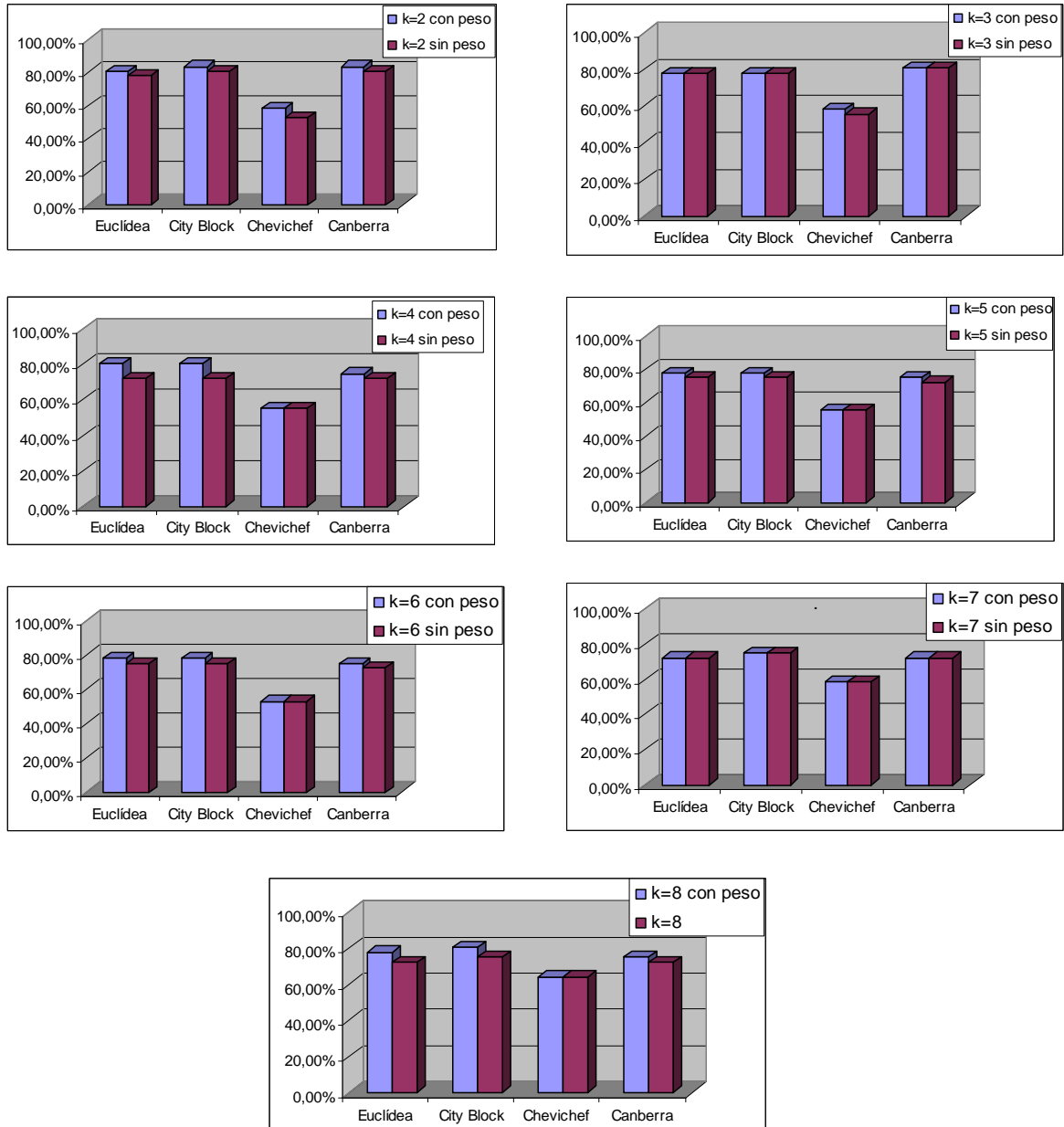


Figura 27: Comparativa entre las diferentes distancias normalizadas con y sin peso para $k \in [2, 8]$.

4.1.4.8 Comparativa entre las diferentes distancias con la propiedad del peso, normalizando y sin normalizar para $k \in [2,8]$

La normalización en los valores de los atributos, no crea un patrón de comportamiento en las diferentes distancias junto con la presencia del peso. De hecho, se puede observar que la mayor parte de los mejores porcentajes se atribuyen a experimentos sin normalización, ya que con este tipo de proceso, se rompe la relación con la posición de llegada de los valores RSSI en la instancia. Todos los valores oscilan entre 0 y 1, como consecuencia las variaciones entre unos y otros son mínimos y se pierde información relevante para el tipo de localización que se estudia, es decir, localización en función de la intensidad de la potencia y la posición de su AP que recibe a un nuevo usuario.

También, aquellas instancias, en las que la normalización sí que es efectiva, se debe a la atenuación que ejerce sobre el efecto de las señales RSSI ruidosas, según los resultados, se pueden suplir con la propiedad del peso.

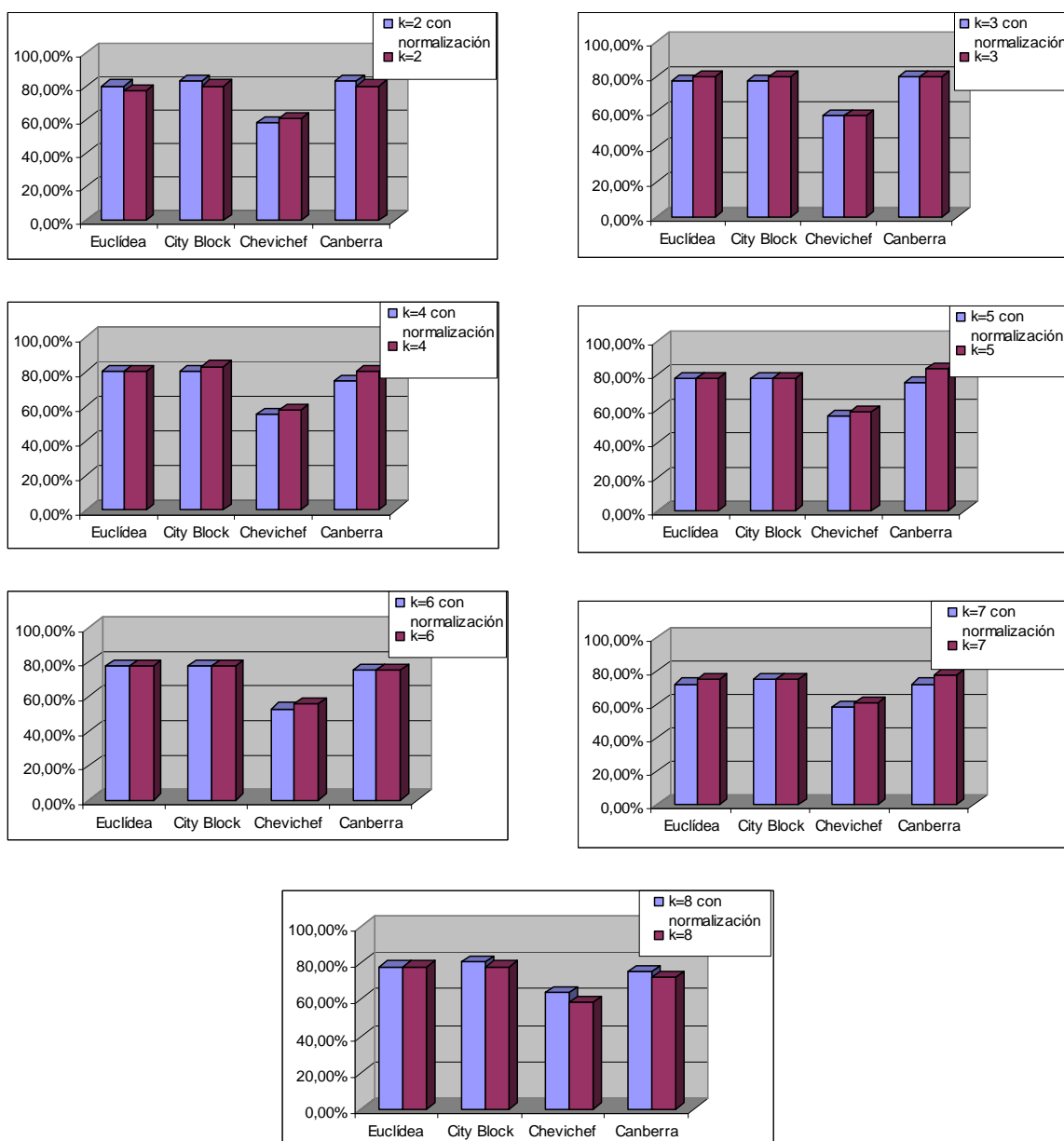


Figura 28: Comparativa entre las diferentes distancias con la propiedad del peso, normalizando y sin normalizar para $k \in [2, 8]$.

4.1.5 Estudio para las diferentes distancias

En este apartado se puede observar el entorno de estudio más beneficioso para cada una de las distancias con las que se trabaja.

4.1.5.1 Resultados obtenidos con la Distancia Euclídea

Como se puede ver en la siguiente gráfica, el número de vecinos que produce mejores resultados en los cuatro tipos de experimentos que se han realizado son $k=3$. Aunque en los que se incluye el peso, el valor par $k=4$ predomina por encima del resto.

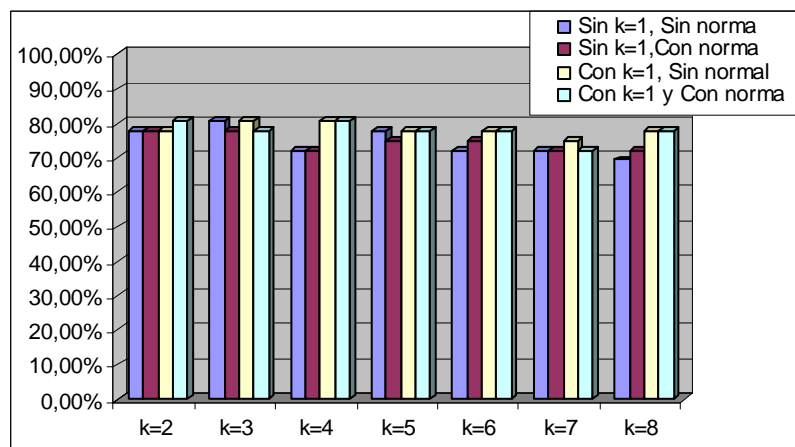


Figura 29: Resultados obtenidos con la Distancia Euclídea.

Otra perspectiva que se puede apreciar es que en la mayor parte de los vecinos más cercanos, obtienen mejores resultados con el experimento en el que se incluye la propiedad del peso y sin normalizar las muestras del entorno, excepto en $k=2$ que la distancia Euclídea, junto con todas las variantes obtiene un 80,56% de acierto.

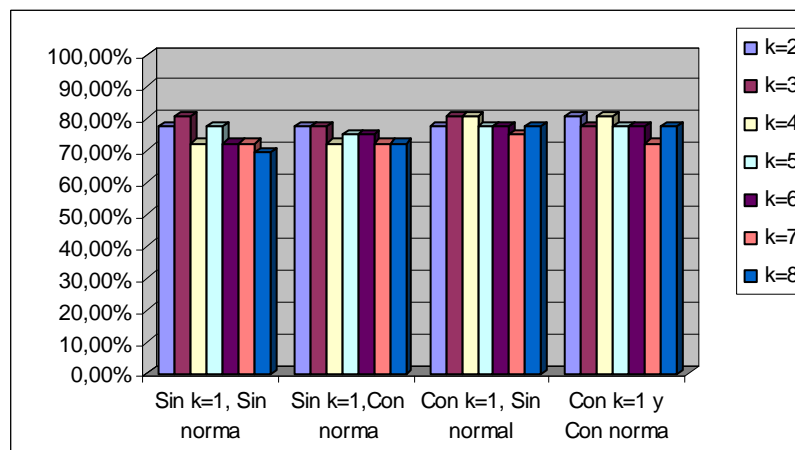


Figura 30: Resultados obtenidos con la Distancia Euclídea.

El resultado óptimo que se ha obtenido con esta distancia es 80,56%.

4.1.5.2 Resultados obtenidos con la distancia City Block

Con la distancia City Block, se puede observar que los mejores resultados se obtienen con los vecinos pares y en concreto con $k=2$.

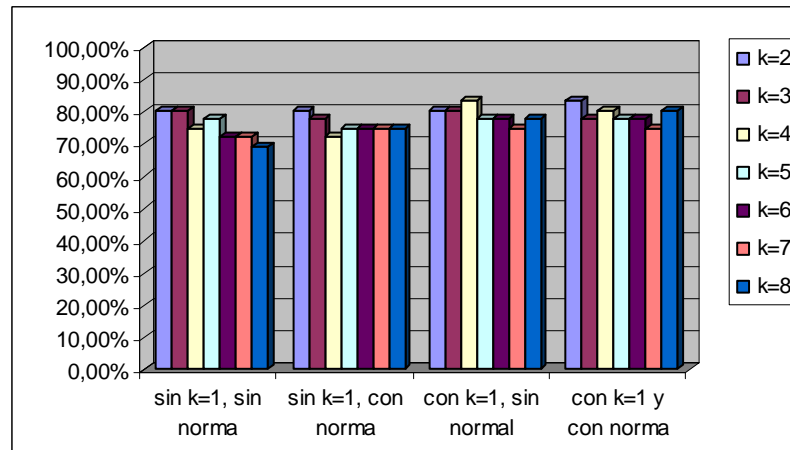


Figura 31: Resultados obtenidos con la Distancia City Block.

Y en la siguiente gráfica, la propiedad del peso es la que permite un mayor número de usuarios localizados correctamente.

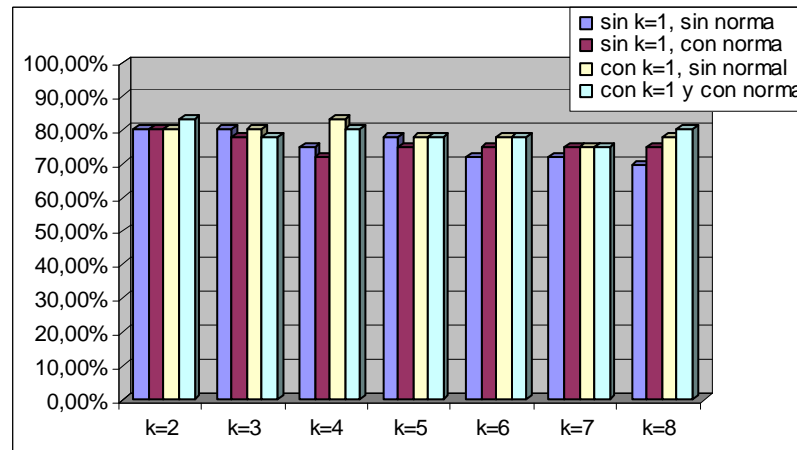


Figura 32: Resultados obtenidos con la Distancia City Block.

El resultado óptimo es 83,3%.

4.1.5.3 Resultados obtenidos con la distancia Chebichef

Los resultados a destacar con la distancia Chebichef, es una vez más la importancia que tiene utilizar el vecino más cercano ($k=1$) en la decisión, una vez que el algoritmo ofrece una indeterminación.

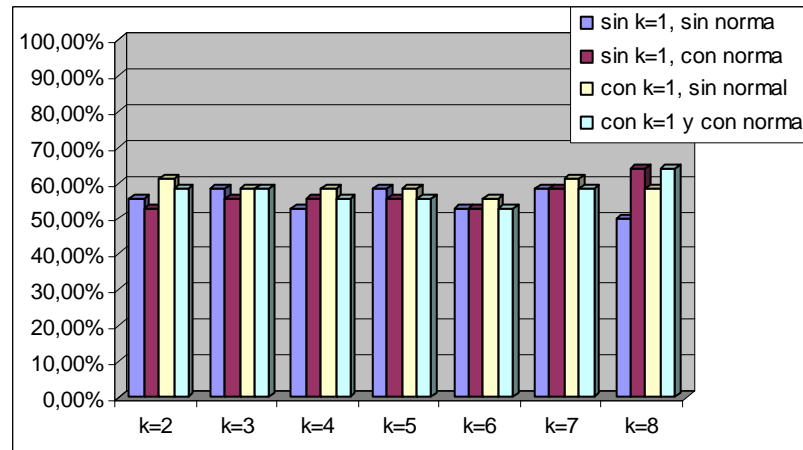


Figura 33: Resultados obtenidos con la Distancia Chebichef.

No se hace una mención especial al valor de k óptimo para la distancia Chebichef, ya que el comportamiento de los mismos es irregular para cada uno de los experimentos, como se puede apreciar en la siguiente gráfica. Aunque se puede destacar $k=8$ con un 63,89%, que es el mejor resultado obtenido con dicha distancia.

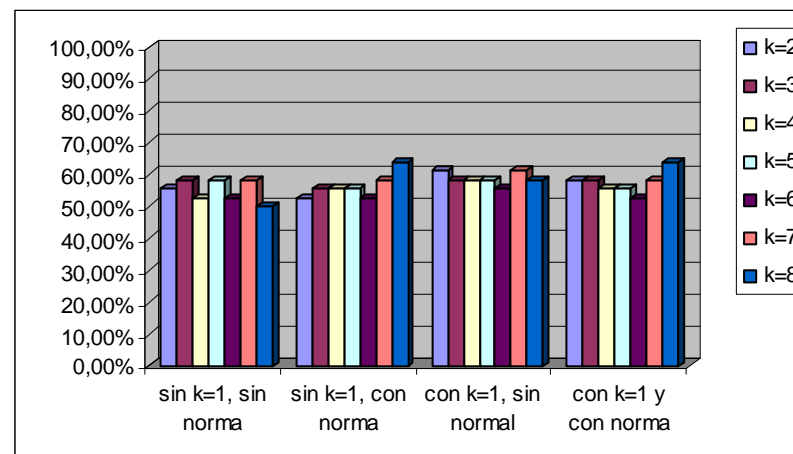


Figura 34: Resultados obtenidos con la Distancia Chebichef.

4.1.5.4 Resultados obtenidos con la distancia Canberra

Los valores de k que obtienen mejores resultados, son valores bajos, es decir, de $k=2$ a $k=5$, y entre ellos el valor más alto 83,3% se corresponde una vez más con el experimento donde se incluye la presencia del peso sin normalizar los datos en el caso $k=5$ y con peso más la normalización de las muestras cuando se decide con dos vecinos.

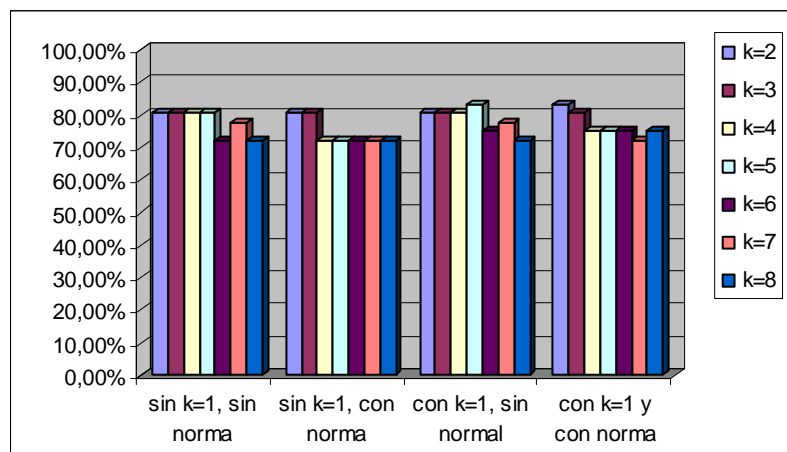


Figura 35: Resultados obtenidos con la Distancia Canberra.

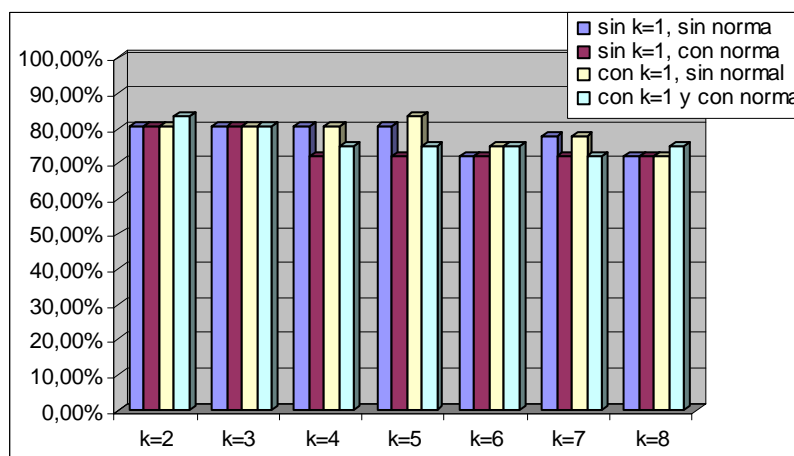


Figura 36: Resultados obtenidos con la Distancia Canberra.

Una vez realizado este análisis, se puede concluir que con un 20% de instancias de entrenamiento, el porcentaje más elevado es un 83,3%, con las distancias City Block y Canberra, con un número de valores de k que oscila entre 2 y 5 y haciéndose imprescindible la presencia de la propiedad del peso para obtener este número de localizaciones correctas.

4.2 FASE II: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training

A continuación, se intenta comprobar si el número de muestras en el grupo de entrenamiento afecta de forma beneficiosa o no en la búsqueda del algoritmo con las condiciones más propicias para la localización. Se ha disminuido el número de muestras de training a un 70% y 60% del total de instancias, por lo tanto, aumentado 30% y 40% de testing respectivamente.

La comparativa que se puede apreciar es la siguiente:

4.2.1 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training N variable, sin peso y sin normalización para $k \in [2, 8]$

Con respecto al número de potencias que forman el vector Fingerprint, los tres grupos se comportan de forma similar, es decir, teniendo en cuenta todos los AP's que se han determinado en la planta, las zonas se identifican más claramente y así mismo ofrece más información para las localizaciones, para cualquier valor de k.

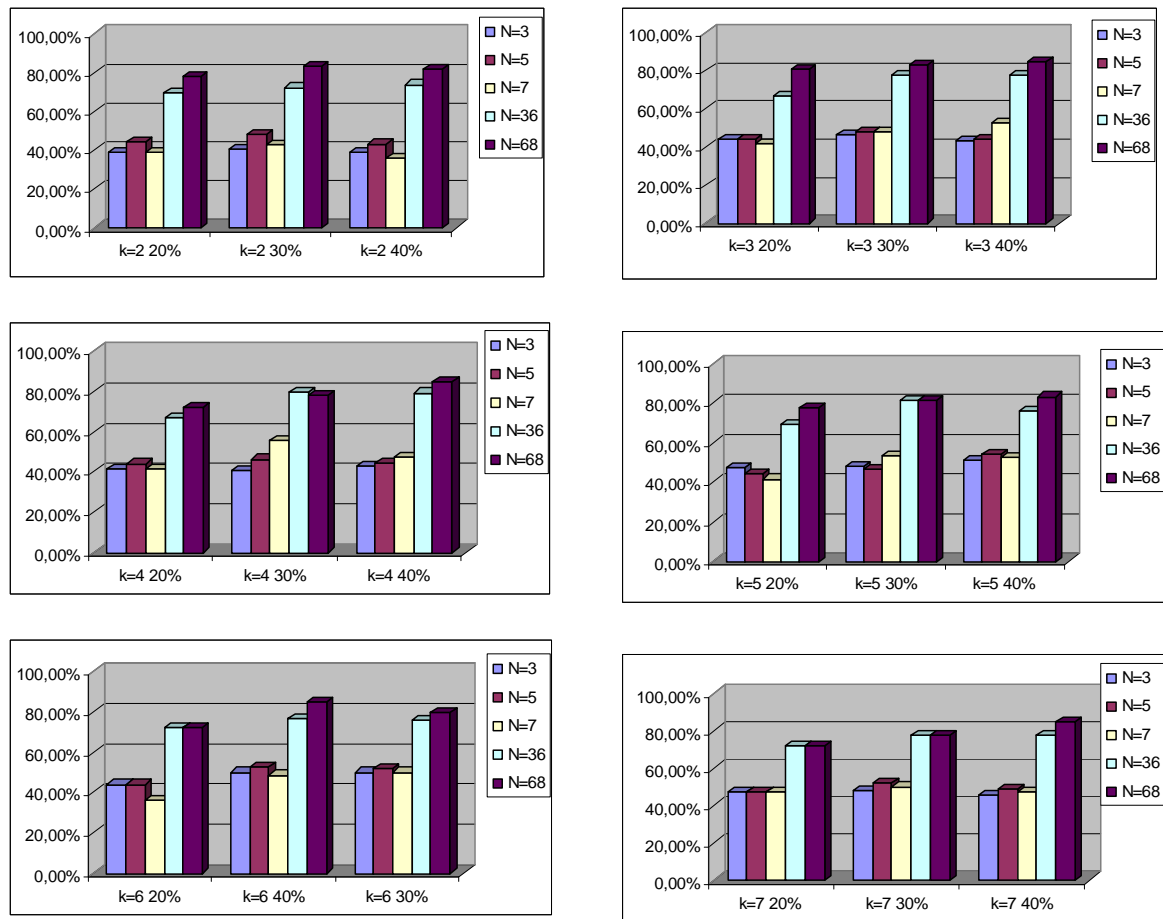
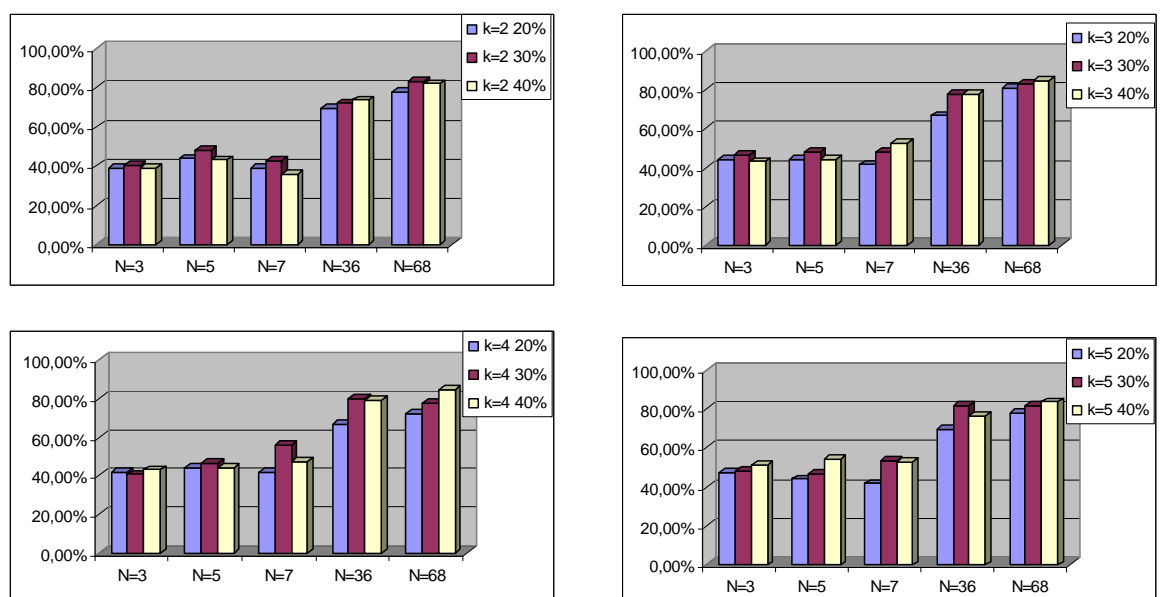


Figura 37: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, sin peso y sin normalización para todo $k \in [2, 7]$.

Para los diferentes valores de N, el grupo de testing que da mejores resultados a modo general es el del 70%, sin embargo el porcentaje óptimo recae sobre el último tamaño del grupo de entrenamiento y como se ha comentado en el apartado anterior con N=68.



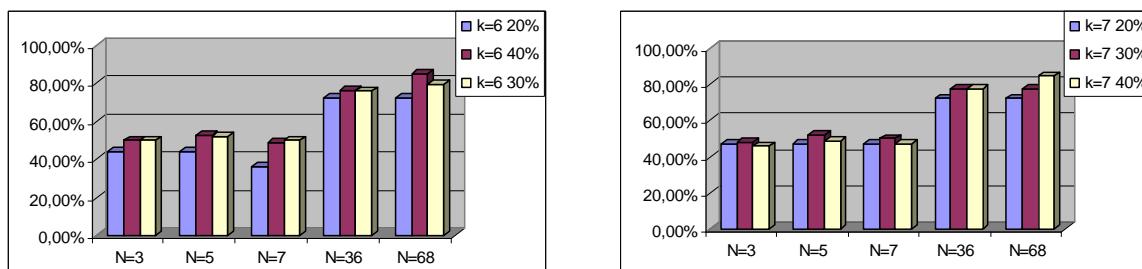


Figura 38: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de training, N variable, sin peso y sin normalización para todo $k \in [2, 7]$.

Se puede introducir una primera conclusión, el 60% de training permite generalizar mejor, eliminando aquellas muestras que o bien ofrecen información irrelevante o que interfieran en las decisiones. Estas muestras se conocen como muestras outlier.

El mejor resultado obtenido es 84,72% para $k=3,4$ y 7 en el grupo de 40% de testing y para $k=6$ en el grupo de 70% de training. A pesar que en ambos conjuntos de entrenamiento se consigue el mejor resultado, se considera como conjunto de instancias más óptimo el del 40% porque el porcentaje más elevado se consigue con un menor número de vecinos que para el grupo del 30%.

4.2.2 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento N variable, sin peso y con normalización para $k \in [2,7]$

Para todo k, los mejores resultados se obtienen con N=68 para los tres experimentos. Se confirma el apartado anterior.

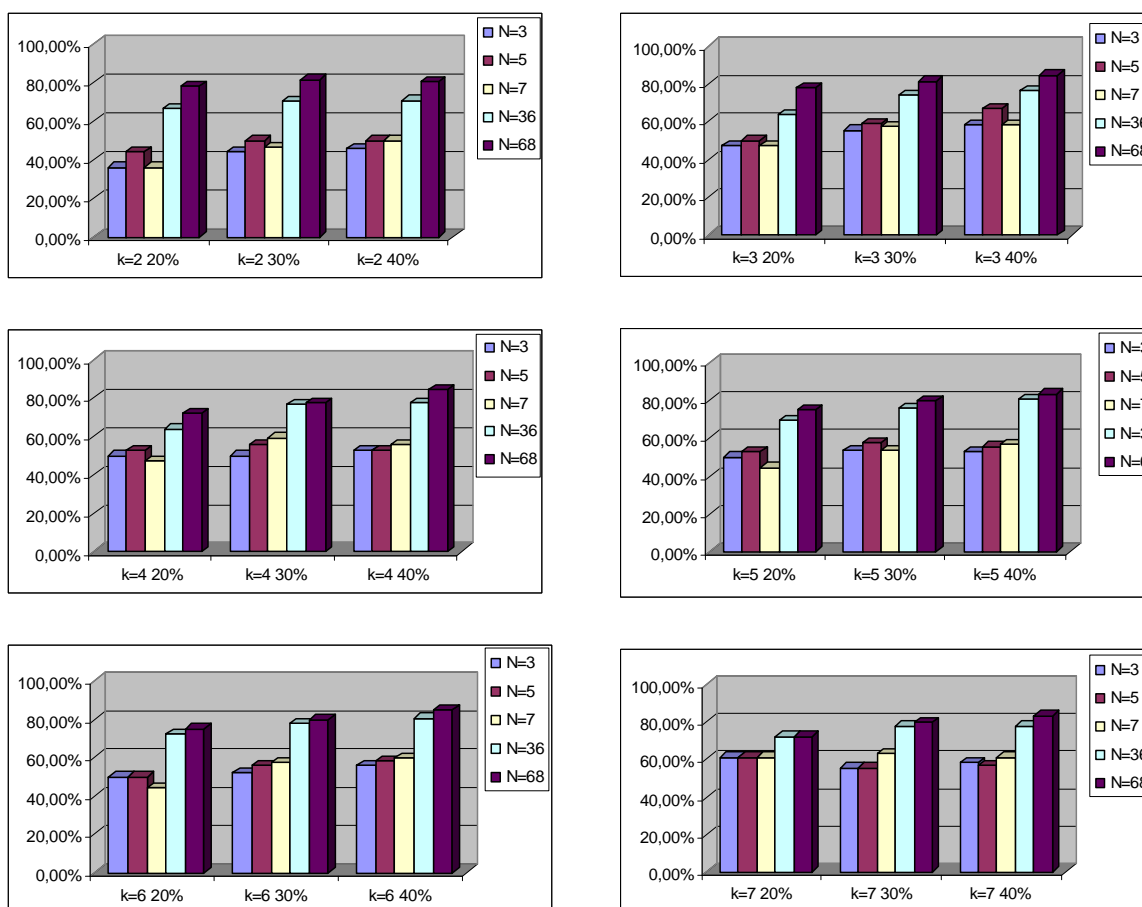


Figura 39: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, sin peso y con normalización para todo k para todo $k \in [2, 7]$.

También se puede decir, que el experimento con un 40% de testing, es el que obtiene los mejores resultados. Destaca $k=3$ y $k=6$ con un 84,7%.

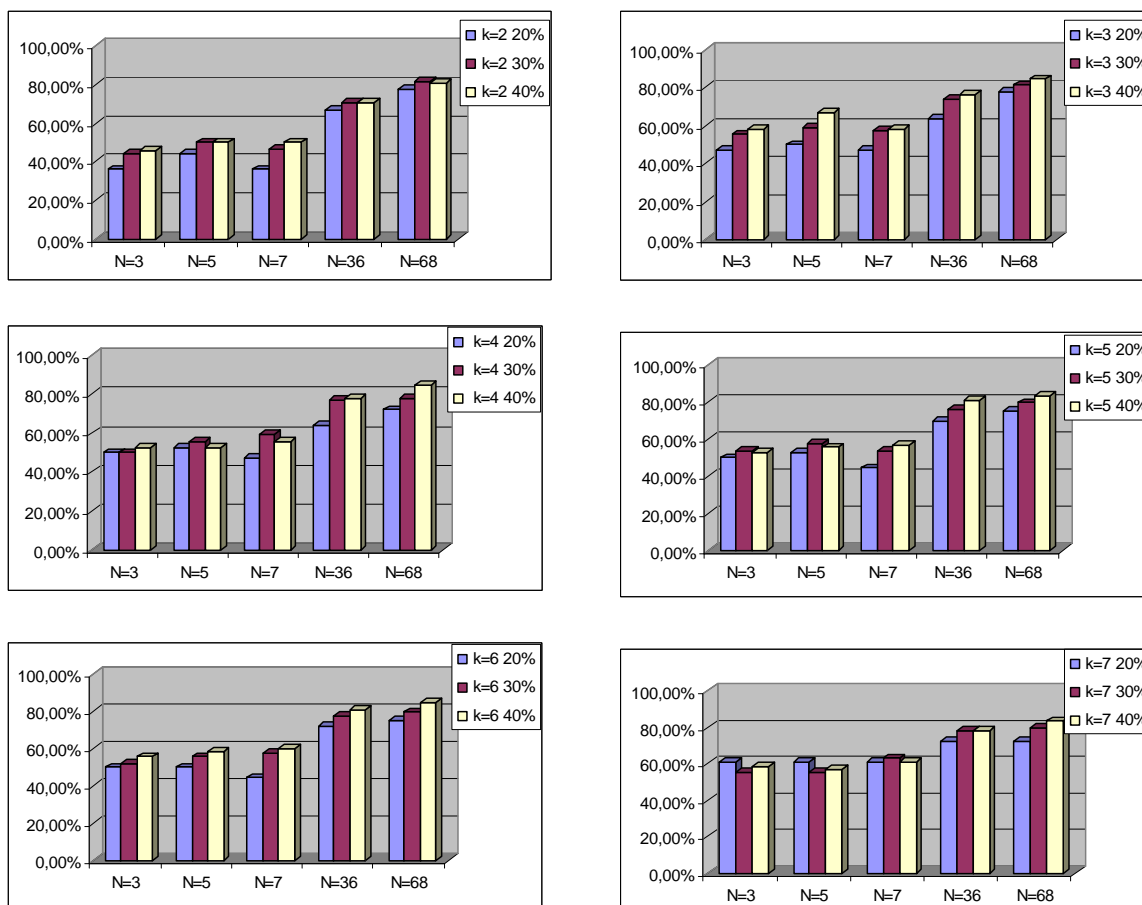
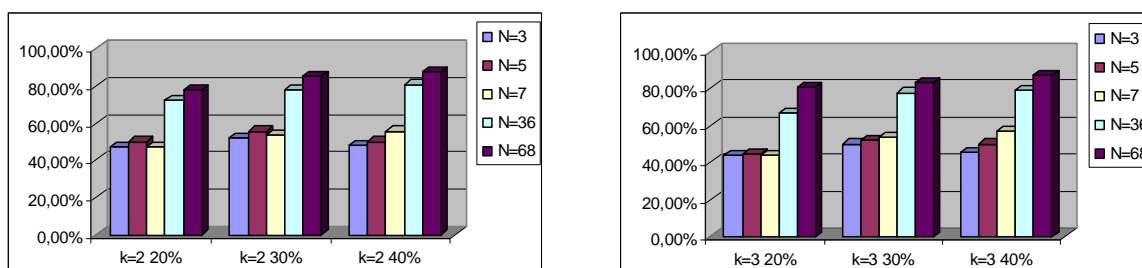


Figura 40: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupos de training, N variable, sin peso y con normalización para todo k para todo $k \in [2, 7]$.

4.2.3 Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento N variable, con peso y sin normalización para $k \in [2, 7]$

Como en anteriores apartados, el mejor resultado se consigue con N=68.



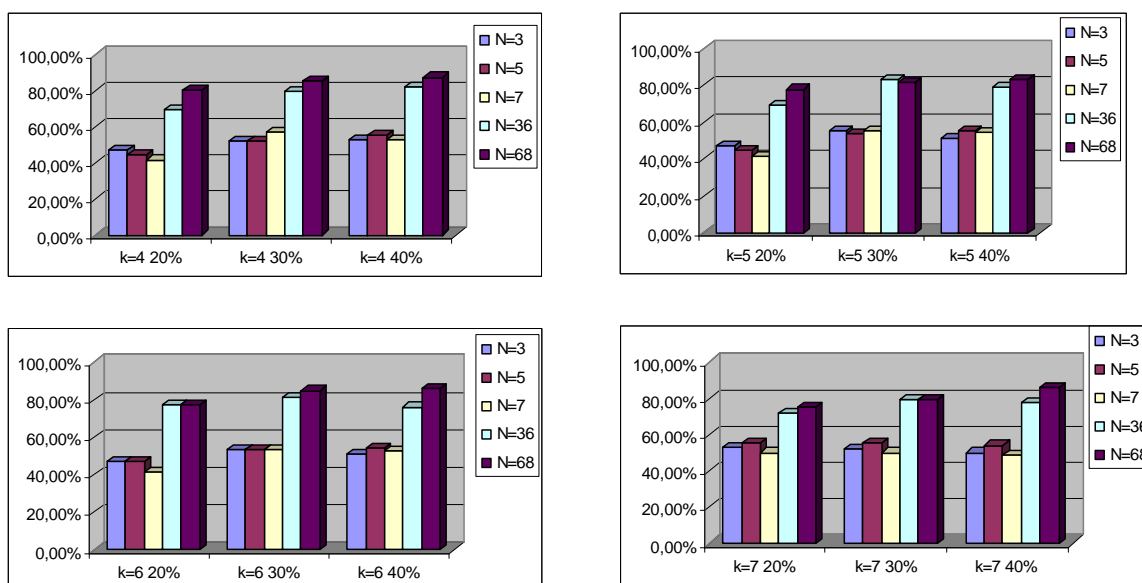


Figura 41: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y sin normalización para todo $k \in [2, 7]$.

Los mejores resultados se dan en el grupo de evaluación de 40% para cualquier k. El valor más elevado es un 88,89% para k=4 y los valores de k más pequeños le siguen con un 87,50% para el mismo grupo de evaluación.

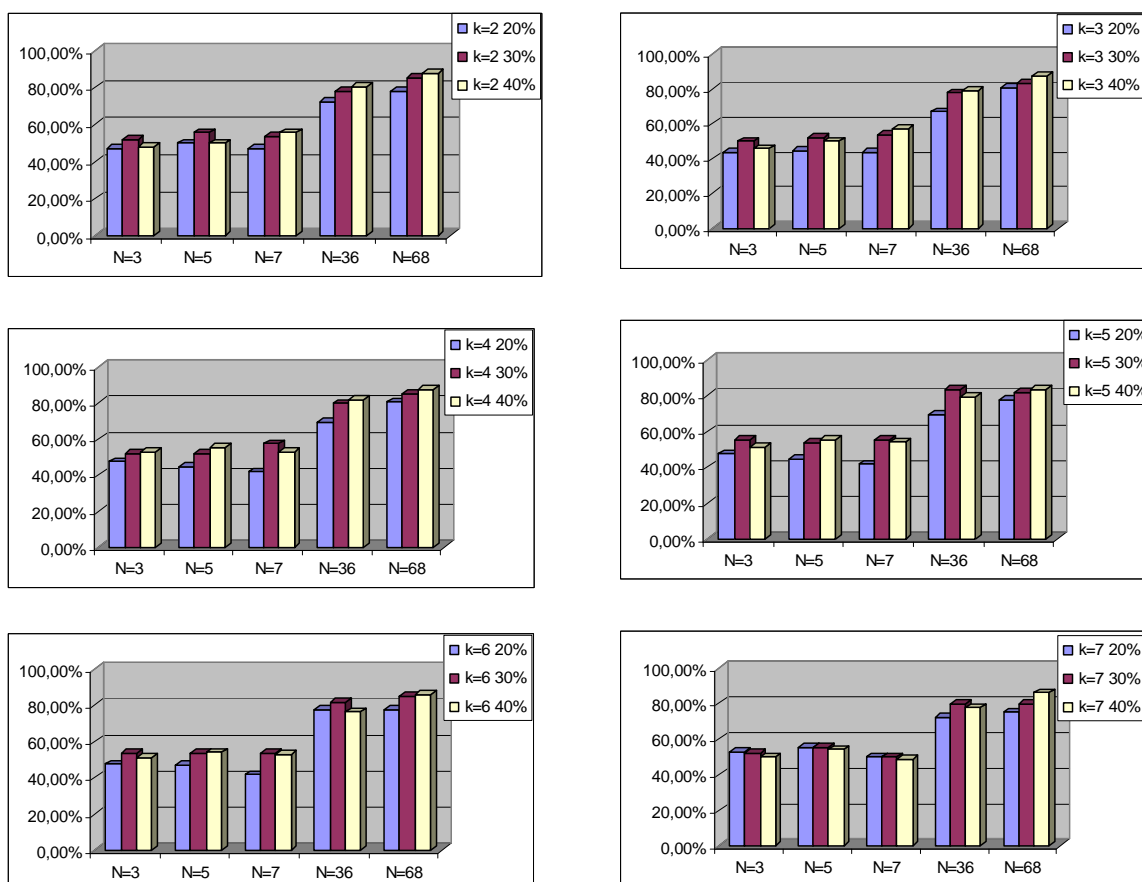


Figura 42: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training, N variable, con peso y sin normalización para k.

4.2.4 Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento N variable, con peso y con normalización para $k \in [2, 7]$

Con respecto al número de AP's que ofrece menor probabilidad de error, es con $N=68$.

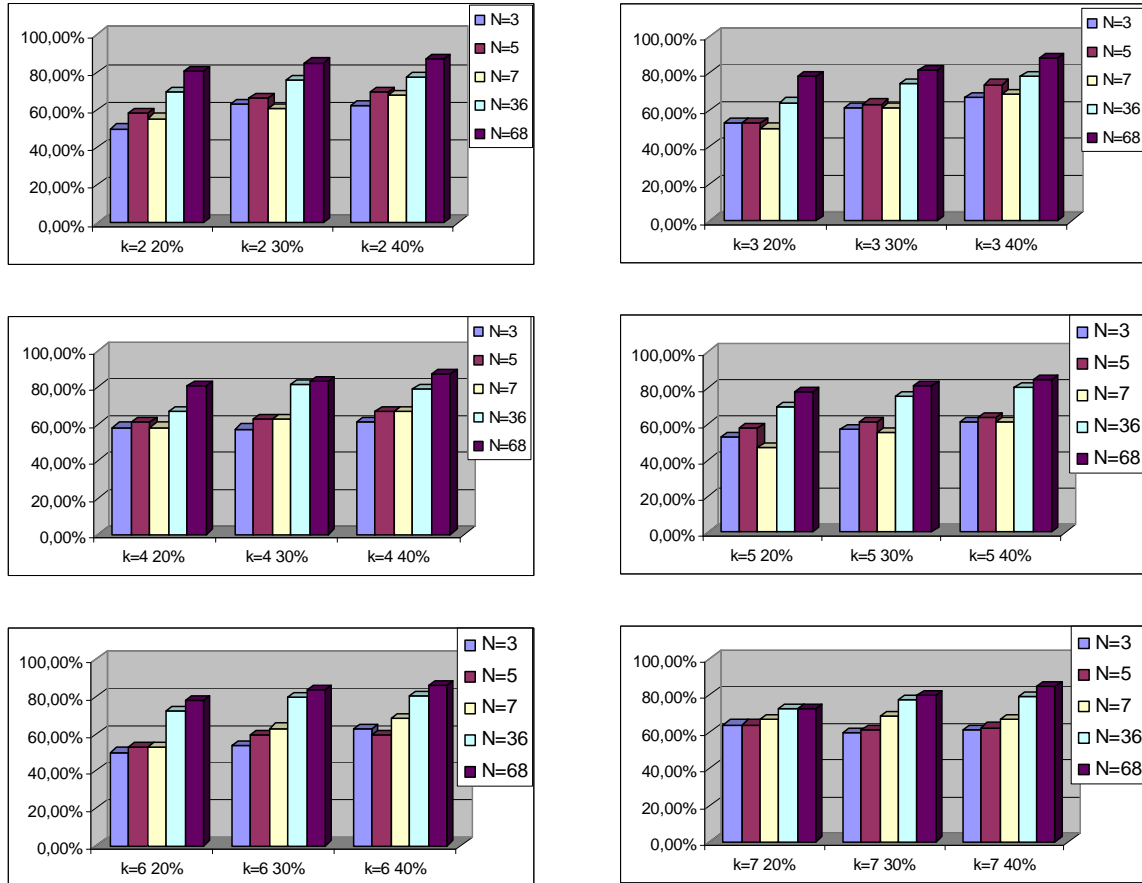
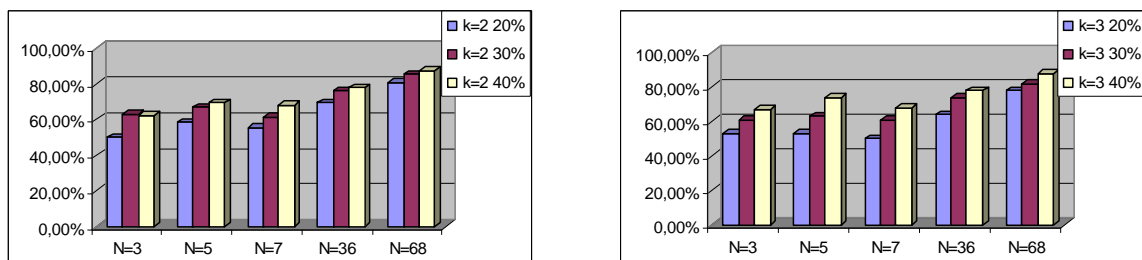


Figura 43: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y con normalización para $k \in [2, 7]$.

El mejor porcentaje de localizaciones exitosas es 87,50% para $k=2, 3$ y 4.



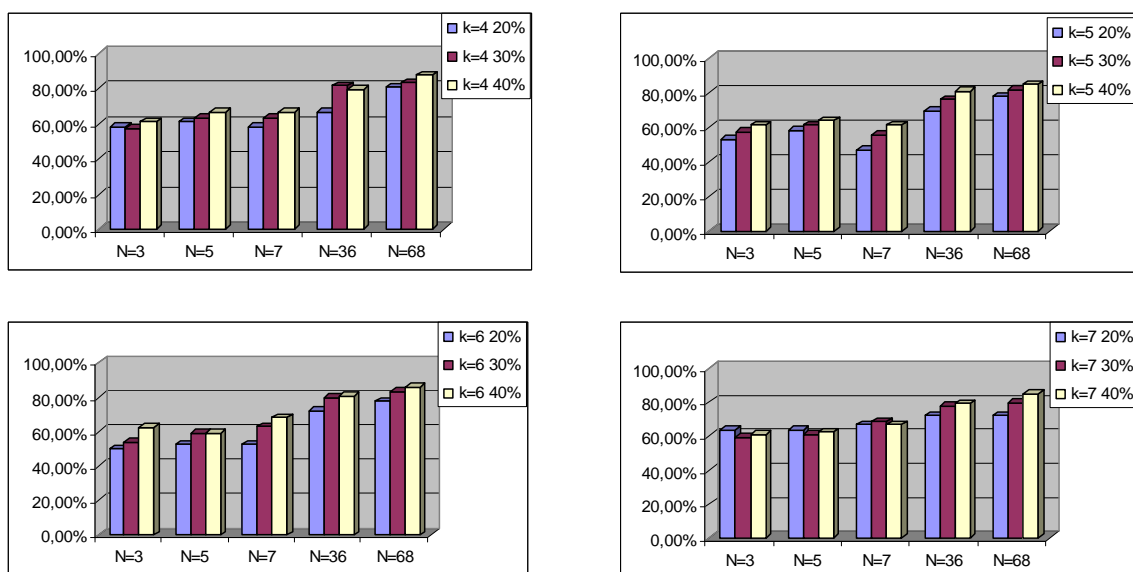
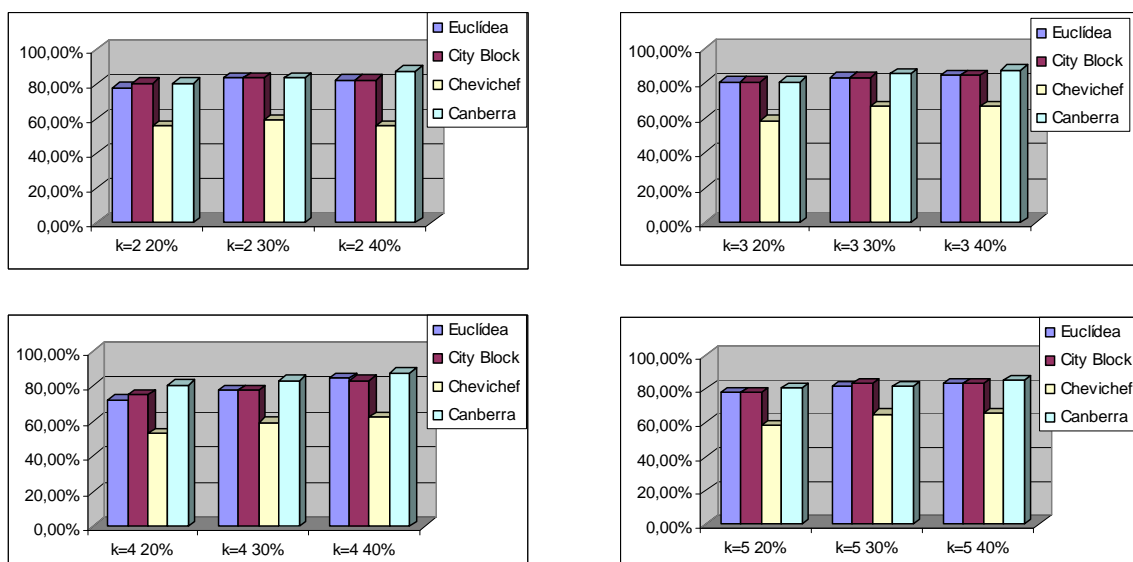


Figura 44: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y con normalización para $k \in [2, 7]$.

A partir de este momento, todos los experimentos se realizarán teniendo en cuenta el número máximo de AP's, ya que como se ha observado en resultados anteriores, N=68 influye positivamente para alcanzar mayor número de localizaciones exitosas.

4.2.5 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y sin normalización para $k \in [2, 8]$

Se puede observar que para todo k, los tres grupos de training se comportan de la misma forma en esta categoría, es decir, Canberra es el que obtiene mejores resultados.



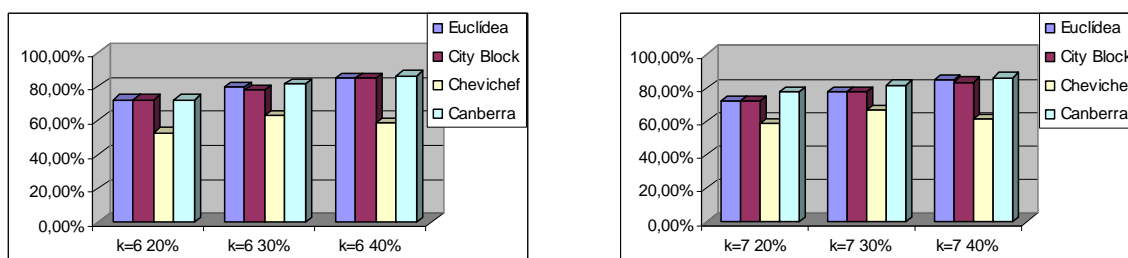
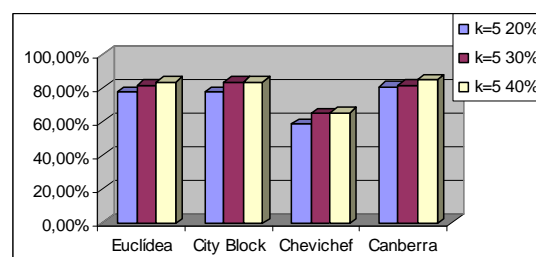
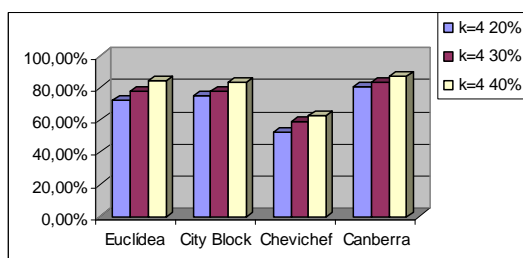
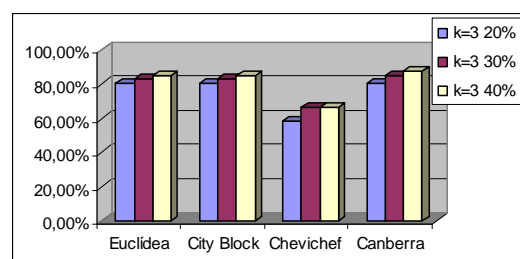
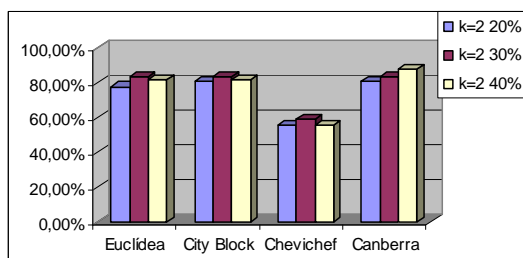


Figura 45: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y sin normalización para $k \in [2, 7]$.

4.2.6 Estudio de las diferentes distancias

- Distancia Euclídea**
 Funciona mejor con el grupo de 40% de testing excepto para $k=2$, que destaca el 30% de testing con un 2% de diferencia con el grupo de 40%.
- Distancia City Block**
 Se comporta igual que la Distancia Euclídea, exceptuando que en $k=5$, con los experimentos de 30% y 40% de testing, se obtienen los mismos resultados.
- Chebichef.**
 Los porcentajes de localización exitosa mejoran al disminuir el número de instancias de training pero no existe una gran diferencia entre 30% y 40% de los mismos conjuntos respectivamente.
- Canberra**
 El grupo que alcanza el porcentaje más elevado de usuarios correctamente localizados es el que se denomina como 40% de testing.



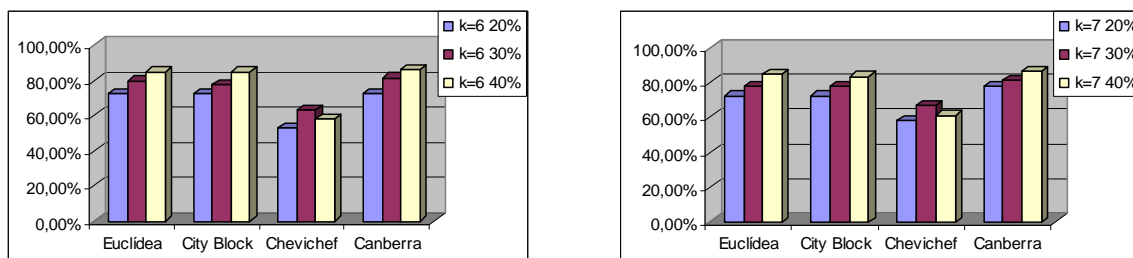


Figura 46: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training y distancias, sin peso y sin normalización para $k \in [2, 7]$.

Al igual que en los experimentos anteriores, el 87,50% corresponde con 60% de training pero con la variante de que la distancia que se ha utilizado en el sistema de localización es Canberra.

4.2.7 Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y con normalización para $k \in [2, 7]$

En general, Canberra continúa siendo la mejor distancia. Se puntualiza que “en general”, porque como se explicó en el análisis individual para el grupo de 20%, la distancia Euclídea y City Block para $k=5, 6$ y 7 superan a Canberra. Sin embargo 30% y 40% tienen un comportamiento homogéneo.

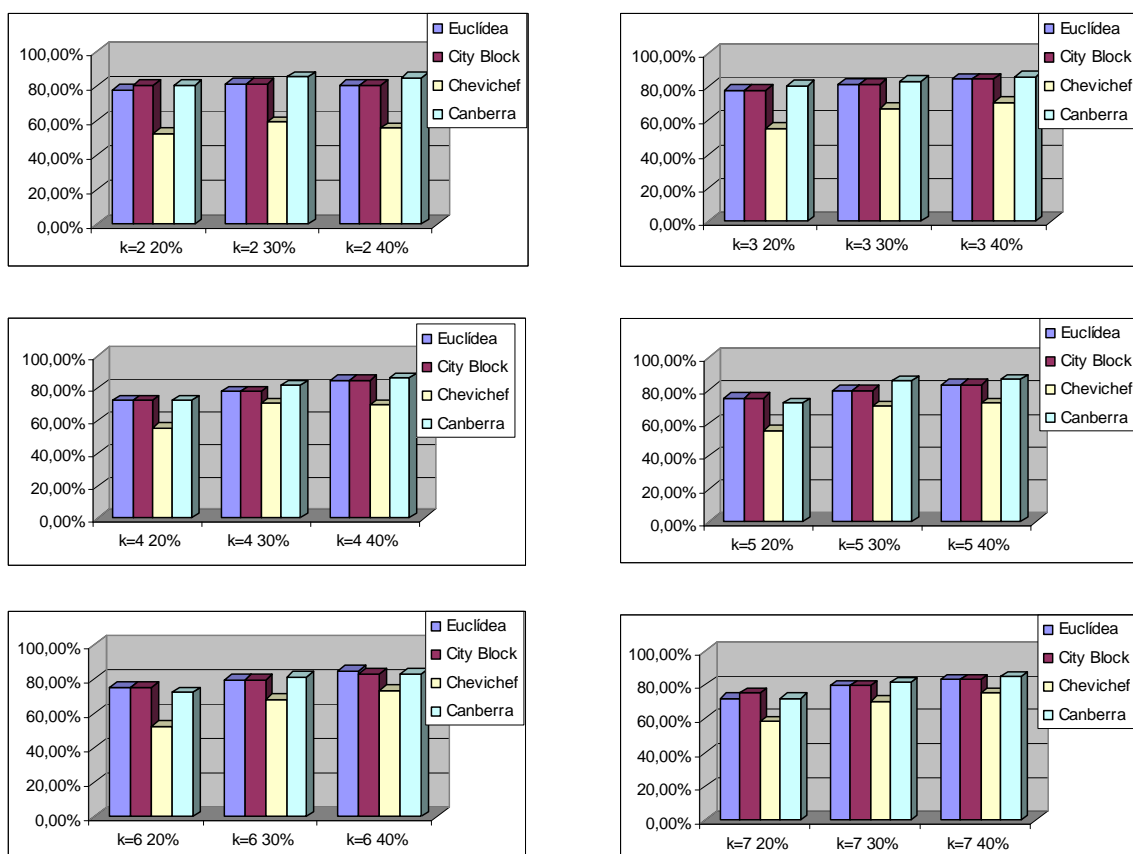


Figura 47: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, sin peso y con normalización para $k \in [2, 7]$.

Al contrario que en el apartado anterior, el grupo de testing del 40% obtiene mayor número de localizaciones para todas las distancias. No hay unanimidad para todo k , pero sí hay mayoría. La diferencia, es la normalización que se ha aplicado en este caso. Se considera que ayuda a que el 60% de

instancias de training sean más generalizadas y por este motivo consigue alcanzar resultados mejores en más entornos diferentes.

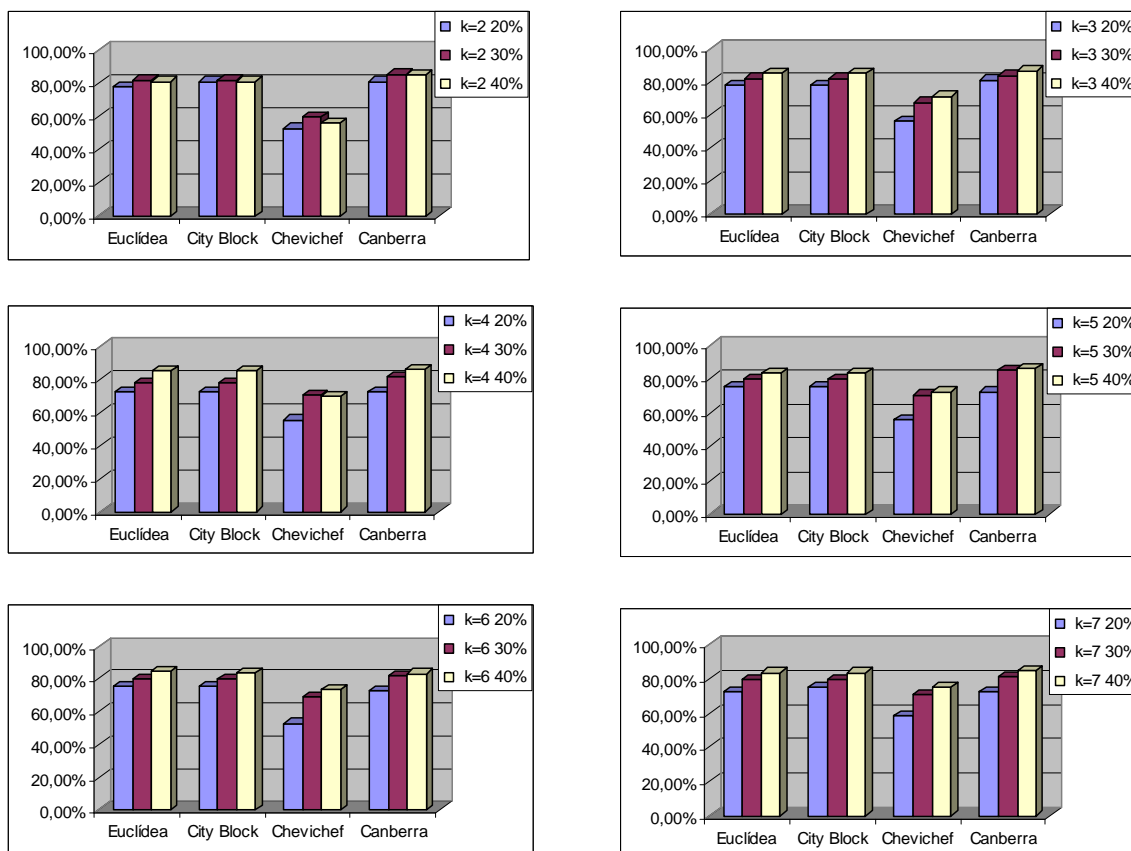


Figura 48: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de entrenamiento y distancias, sin peso y con normalización para $k \in [2, 7]$.

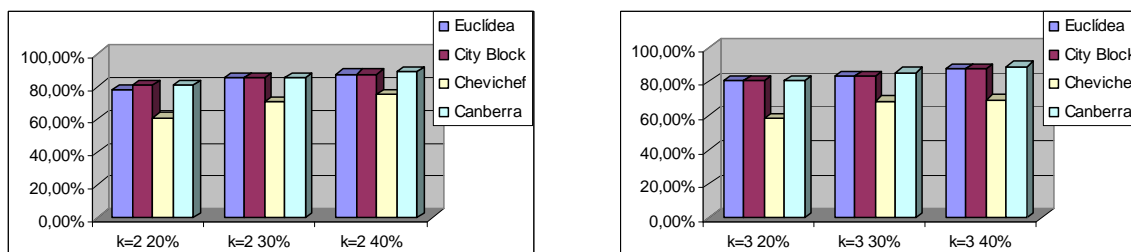
Por último, se indica que el mejor resultado es un 86,10% que se atribuye a la distancia Canberra con $k=3, 4$ y 5 .

4.2.8 Comparativa del comportamiento del algoritmo con diferentes tamaños de conjunto de entrenamiento y distancias, con peso y sin normalización para $k \in [2, 7]$

En esta situación, existe diferencia de distancias predominante, en función de si el número de vecinos elegido es par o impar.

Para los k pares, en los conjuntos de 30% y 40%, la distancia Euclídea y CityBlock es muy parecida a Canberra o incluso superior en $k=6$. Para el 20%, la distancia CityBlock es la que registra valores óptimos. Sin embargo con k impares, la distancia Canberra, es superior o igual a la distancia Euclídea y City Block para los tres experimentos.

Vuelve a cobrar fuerza la presencia del peso en los tres conjuntos de experimentación.



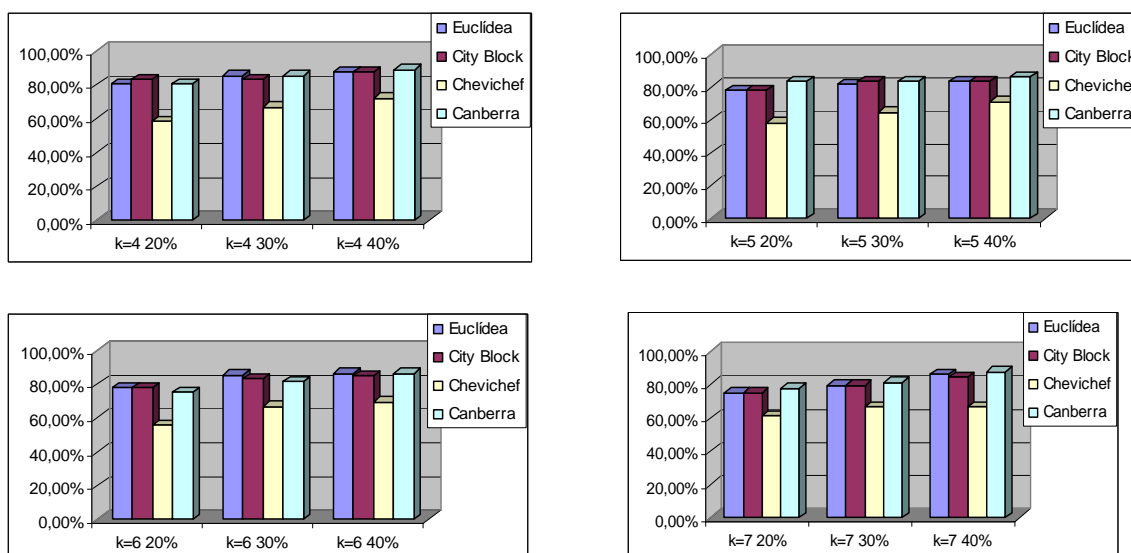


Figura 49: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de training y distancias, con peso y sin normalización para $k \in [2, 7]$.

El mejor resultado que se obtiene es un 88,89% con la distancia Canberra para $k=2, 3$ y 4 y este último también la distancia Euclídea. Se puede deducir que Canberra es la que mejor se comporta para este sistema de localización, ya que con menos vecinos y como consecuencia menos carga computacional, alcanza el porcentaje óptimo hasta el momento.

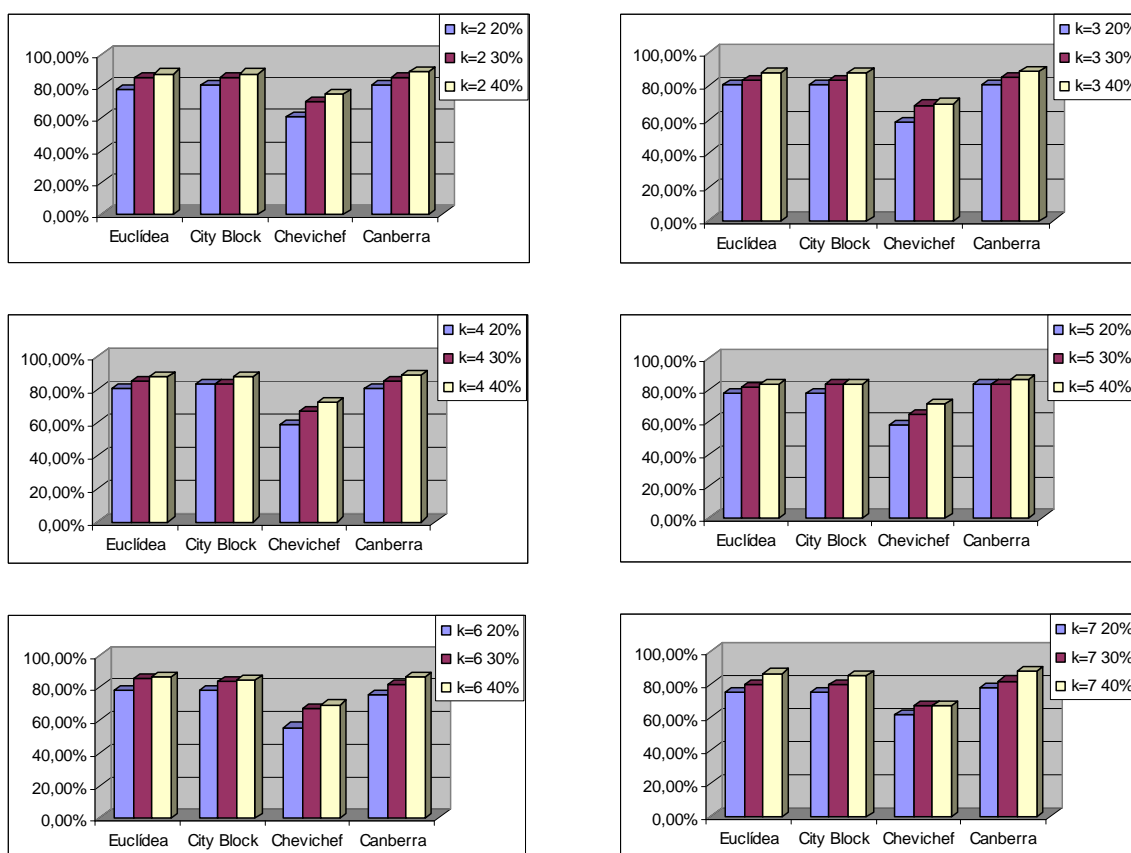


Figura 50: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, con peso y sin normalización para $k \in [2, 7]$.

4.2.9 Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de training y distancias, con peso y con normalización para $k \in [2, 8]$

La normalización que se aplica sobre las instancias del conjunto de entrenamiento, permiten que las distancias Euclídea y CityBlock aumenten su porcentaje de éxito, alcanzando a la distancia Canberra.

Los tres grupos actúan de la misma forma ante las variaciones que se introducen en el experimento. El resultado óptimo se identifica con un 88,89% de localizaciones para $k=4$ con Canberra. El resultado se consigue con números de vecinos pares, gracias a la presencia de la propiedad del peso.

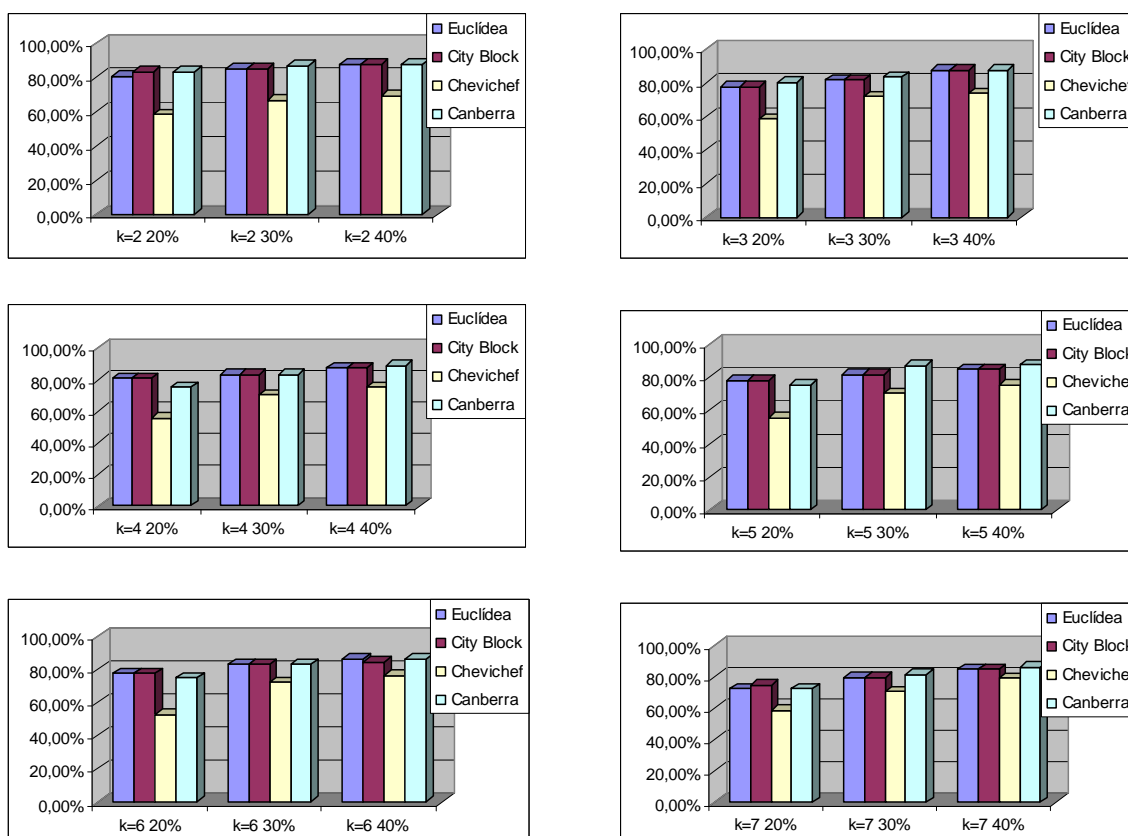
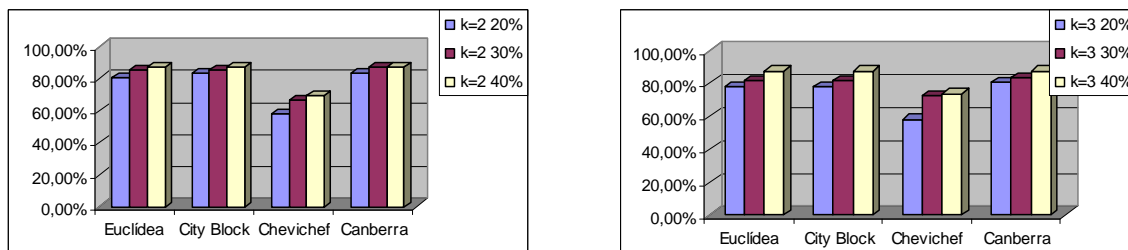


Figura 51: Comparativa del comportamiento del algoritmo con diferentes tamaños de conjuntos de entrenamiento y distancias, con peso y con normalización para $k \in [2, 7]$.

Según el desarrollo de los experimentos, se puede confirmar la premisa que se enunció al principio de este apartado, es decir, a medida que ha disminuido el número de instancias en el conjunto de entrenamiento, los resultados han sido más satisfactorios. Se puede explicar este hecho, gracias a que se han eliminado instancias que influían negativamente en las decisiones del algoritmo, muestras outlier, y como consecuencia ha aumentado su capacidad de generalización, en detrimento a la posibilidad de identificar muestras con características puntuales, que es uno de los objetivos que se quiere alcanzar.



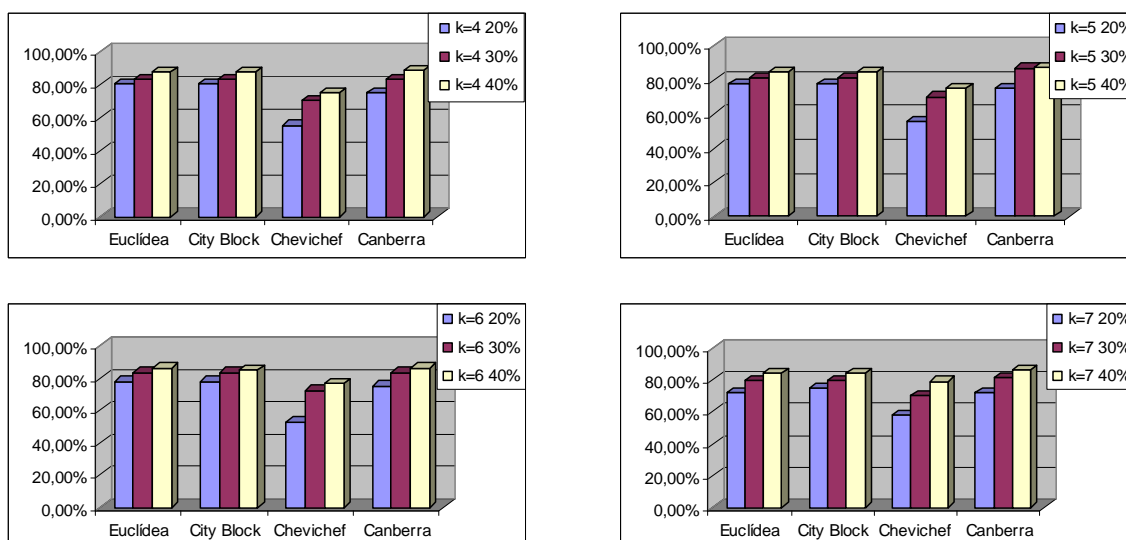


Figura 52: Comparativa del comportamiento del algoritmo con diferentes tamaños de grupo de entrenamiento y distancias, con peso y con normalización para $k \in [2, 7]$.

4.3 FASE III: Optimización del algoritmo

Teniendo en cuenta la posibilidad de disminuir el coste y carga computacional, se considera que el experimento, hasta el momento es el que consigue un 11,11% de tasa de error, sin normalizar los datos, aplicando la medida de similitud a los 68 AP's distribuidos por la planta del Torre Quevedo, más la presencia del peso en las distancias de los $k=2$ vecinos como resultado de la distancia Canberra.

El siguiente objetivo es optimizar el algoritmo en dos puntos:

- **Carga computacional.**
- **Filtrado de ruido.**

El primer subapartado intentará alcanzar el mismo porcentaje exitoso, pero con un proceso de ejecución más rápido. El segundo apartado, explica cómo eliminar las instancias ruidosas que impiden que el sistema sea más resolutivo.

Estas dos últimas variantes se aplican en el experimento con las características que se han determinado como óptimas y han alcanzado un 88,89% de localizaciones satisfactorias.

4.3.1 Carga computacional

La aplicación del método incremental k-dd, ofrece los mismos resultados, es decir, el porcentaje de éxito es 88,89%, que sin su presencia en la ejecución del sistema. La diferencia, es que no aparecen decisiones indeterminadas, incluso con número de vecinos pares como $k=2$ y 4 que son valores que tienen más posibilidad de no llegar a una localización clara.

El uso del método denominado dinámico, sustituye la propiedad del peso. Se puede deducir, que en aquellas situaciones donde se producía una indeterminación, el resultado estaba notablemente influido por la presencia de instancias RSSI que provenían de zonas no contiguas al área donde se encuentra el nuevo usuario a localizar, ya que el método dinámico, obtiene la decisión, sólo a partir de aquellas instancias que pertenezcan a zonas contiguas del área de cobertura donde se encuentre el dispositivo móvil a localizar.

A continuación se presenta el volcado de información que se obtiene a partir del proceso del sistema de localización.

dinámico limita este tipo de instancias ruidosas producidas por el efecto del multipath y en conclusión, la decisión se obtiene a partir de dos vecinos de la misma zona y en este ejemplo en concreto, correcta.

Se ha comentado, que la aplicación del método dinámico, también permitía, reducir el tiempo que tardaba el sistema de localización en obtener una respuesta. Habría que puntualizar que el tiempo que disminuye es el del proceso de tomar una decisión, ya que hay menos instancias sobre las que aplicar la distancia a elegir. Sin embargo, en global, la duración es mayor, debido a la obtención del subconjunto de entrenamiento a partir del grupo de training original.

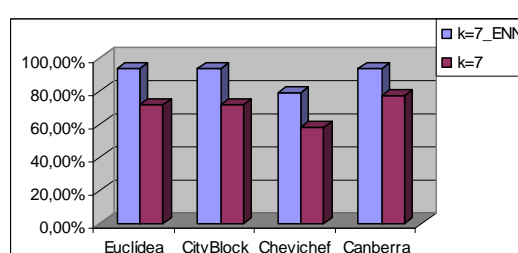
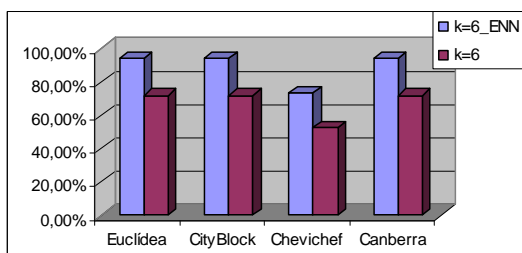
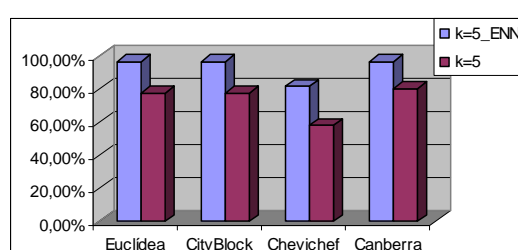
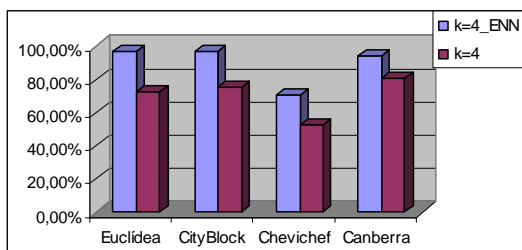
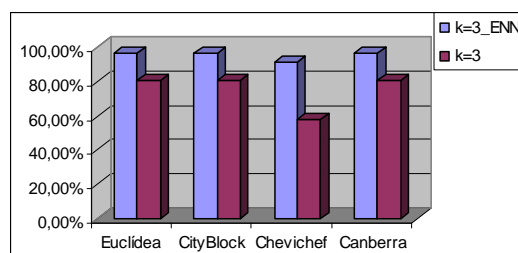
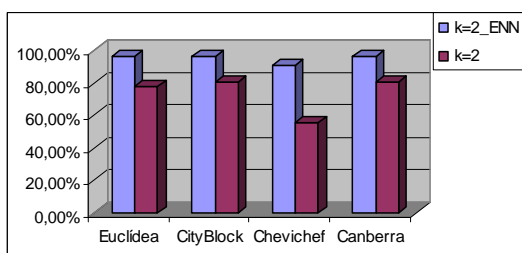
4.3.2 Filtrado de ruido

En este punto del proyecto, se intenta construir un nuevo subconjunto de entrenamiento a partir del original, que está formado por las instancias de éste que han sido localizadas correctamente por las propias instancias del conjunto del Fingerprint al que pertenecen.

El primer experimento que se ha realizado, ha consistido en formar un subconjunto del conjunto de entrenamiento que corresponde con el 80% de instancias del grupo de training. Para ello se ha aplicado el algoritmo k-NN con el máximo número de potencias a examinar en un Fingerprint, es decir, $N=68$, distancia Euclídea como medida de diferenciación y con un número de vecinos más cercanos $k=3$.

Una vez aplicado el sistema, se obtuvieron diez instancias que no habían sido correctamente localizadas o cuya localización era indeterminada, como consecuencia el nuevo conjunto de training estará formado por las instancias originales pero sin la presencia de estos vectores, que se pueden considerar como ruido. A continuación, se repiten las pruebas que se habían realizado con k-NN básico. Se puede observar que el sistema de localización mejora su capacidad de acierto, para todo k, con cualquier medida de similitud y bajo la condición de normalización y peso y sin la presencia de los mismos, como se puede ver en las gráficas siguientes.

4.3.2.1 Comparativa entre KNN y ENN con k variable, sin peso y sin normalización de datos



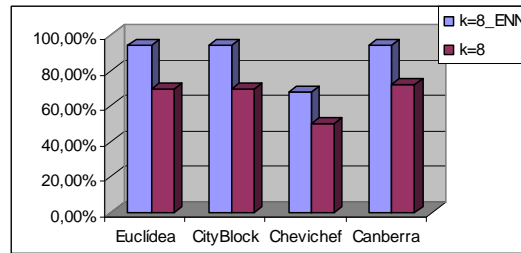


Figura 53: Comparativa entre KNN y ENN con k variable, sin peso y sin normalización de datos.

4.3.2.2 Comparativa entre KNN y ENN con k variable, con peso y sin normalización de datos

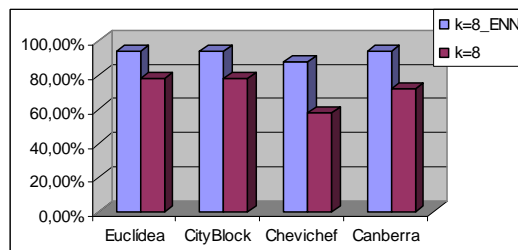
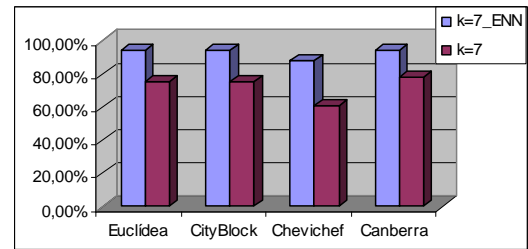
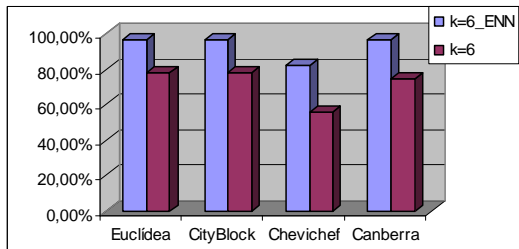
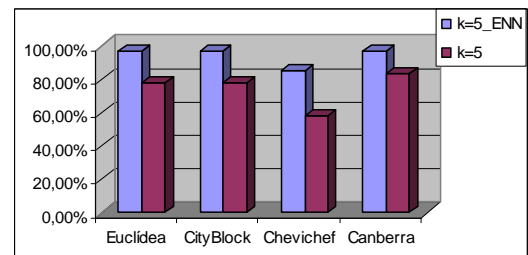
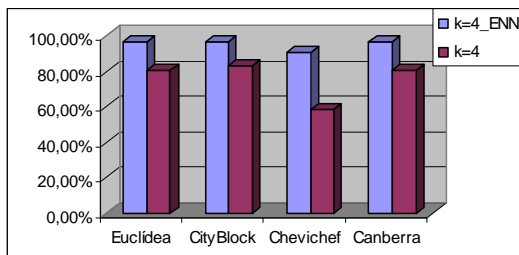
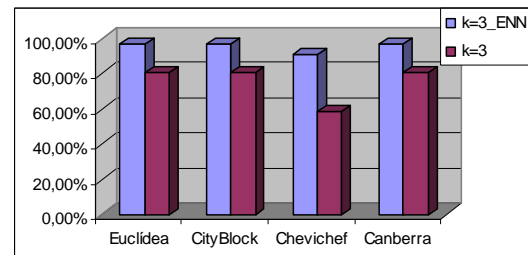
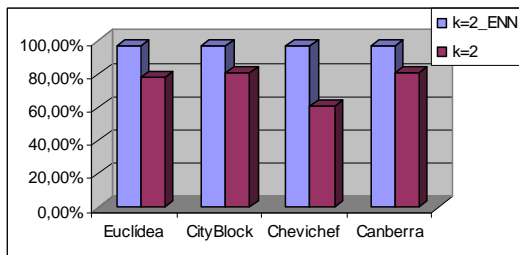


Figura 54: Comparativa entre KNN y ENN con k variable, con peso y sin normalización de datos.

4.3.2.3 Comparativa entre KNN y ENN con k variable, sin peso y con normalización de datos

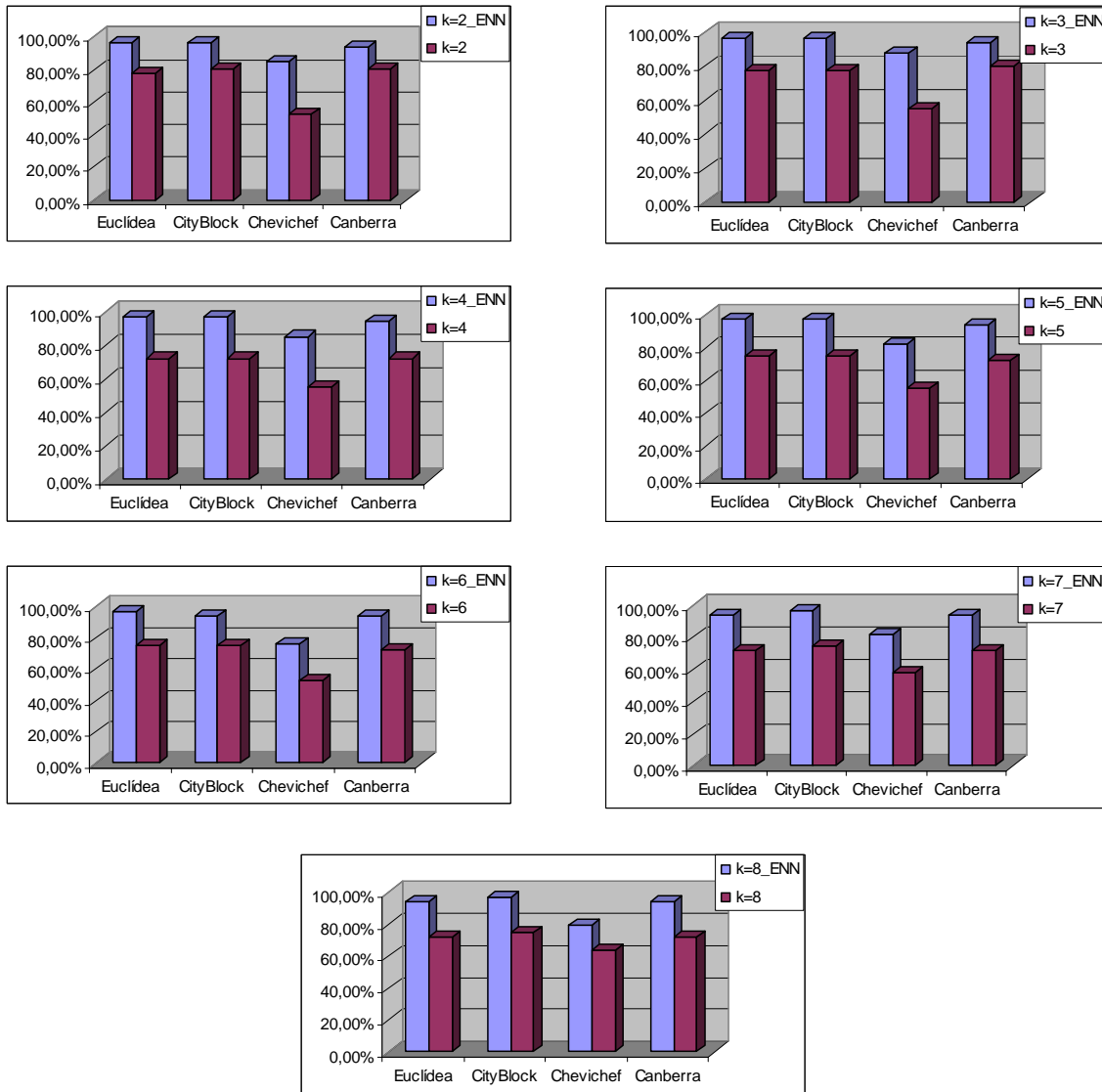
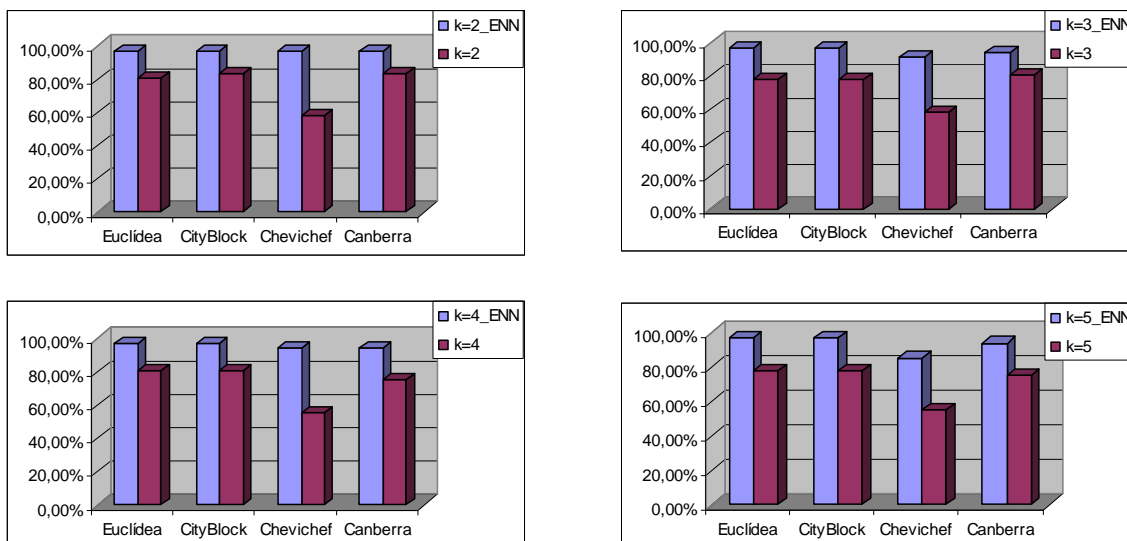


Figura 55: Comparativa entre KNN y ENN con k variable, sin peso y con normalización de datos.

4.3.2.4 Comparativa entre KNN y ENN con k variable, con peso y con normalización de datos



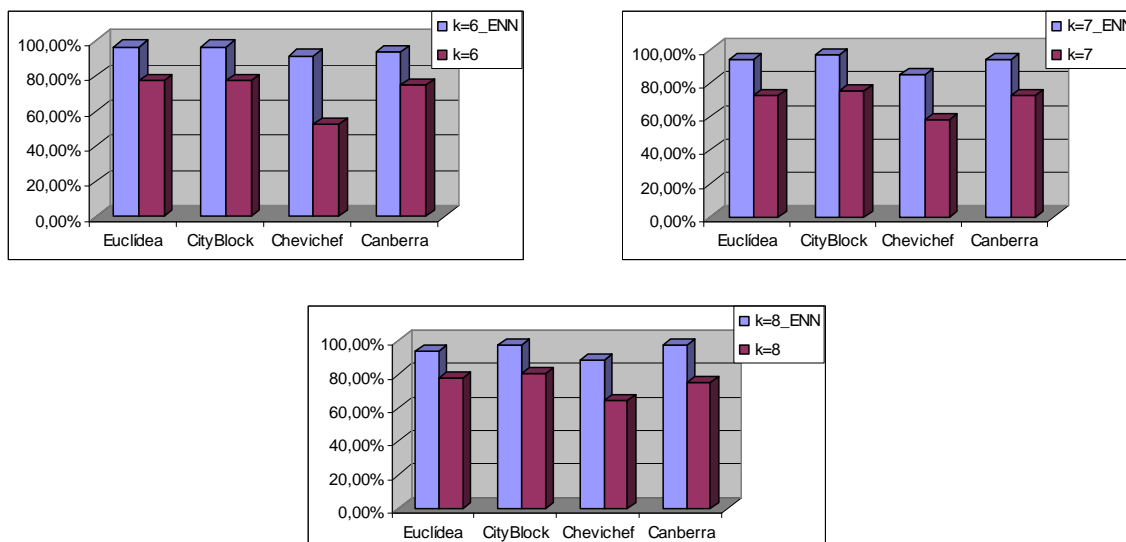


Figura 56: Comparativa entre KNN y ENN con k variable, con peso y con normalización de datos.

En cada uno de los experimentos, el porcentaje de éxito aumenta hasta alcanzar el valor más alto con un 97%.

El mismo proceso se realizó con el grupo de entrenamiento de 70% y 60%. Los resultados fueron satisfactorios, es decir, el número de localizaciones correctas superó a los respectivos resultados de k-NN básico y dinámico. Estos resultados confirman la finalidad del algoritmo ENN, es decir, eliminar las instancias de training ruidosas que influyen negativamente en la obtención de localizaciones correctas.

Si se comparan los porcentajes de éxito obtenidos en las tres variantes, se puede observar que los mejores resultados se obtienen con el subconjunto de entrenamiento más numeroso. Se puede deducir que una vez filtrado, cuanto mayor número de instancias formen el grupo de entrenamiento, más puntos cubren una misma zona de cobertura y más posibilidad existe que la instancia de un usuario a localizar coincida o tome medidas muy similares a dichas medidas.

4.4 FASE IV: Resumen de resultados

Se puede concluir que el algoritmo ENN, con la distancia Euclídea y un número de vecinos reducidos, como $k=3$, permite un sistema de localización potente con respecto al porcentaje de éxito de ubicaciones, sin tener en cuenta el coste computacional. En el caso, que la fase de training, tenga que ser lo más corta posible, la distancia Canberra, junto con un número de vecinos no muy numeroso, obtienen resultados aceptables.

5.- CONCLUSIONES Y TRABAJOS FUTUROS

A partir de los resultados que se han obtenido en función de los experimentos realizados, se pueden obtener las siguientes conclusiones que están directamente relacionadas con los objetivos que se habían planteado en la introducción.

5.1 PARÁMETROS

- Considerar todos los AP's que están presentes en el entorno de localización ayuda a obtener mejores resultados. En el caso de estudio, cada zona se encuentra determinada por un máximo de 12 direcciones MAC, que varían en función del área de cobertura.

La posición fija que ocupan los AP's en el Fingerprint de cada instancia es lo que permite determinar la ubicación. Como consecuencia, si se consideran todos los dispositivos, en todas las instancias siempre se encuentran presentes las que identifican la zona donde se encuentra el nuevo individuo, de esta forma hay más probabilidad de acierto.

- El número de vecinos más cercanos que ofrece resultados óptimos es $k=3$ y $k=4$. Como se puede observar es un grupo reducido de instancias, pero a medida que se aumenta, se comprueba que la distancia mínima también y por este motivo más posibilidad existe de coger vecinos frontera de otras regiones y el algoritmo se vuelve más sensible al ruido.
- El algoritmo k-NN básico aumenta su capacidad de generalización, si el conjunto de entrenamiento no es excesivamente numeroso, ya que cuantas más instancias se tomen, más probabilidad existe de que se incluyan instancias que podrían hacer referencia a muestras outlier, efecto del multipath u obstáculos en el momento de la toma de muestras, que no recogen el comportamiento normal de una instancia.

5.2 ATRIBUTOS

- La presencia del peso permite resolver la indeterminación de una localización en caso de ambigüedad, tomando como solución la distancia mínima a una nueva instancia, es decir, $k=1$. Se parte del hecho que el vecino más cercano es el que más cerca se encuentra al nuevo usuario y como consecuencia, en una localización similar. Al tomar esta decisión se elude la presencia del multipath, de ahí que algunos resultados no sean satisfactorios.
- La normalización es exitosa si se aplica durante la diferencia de potencias RSSI, no en los datos originales de las instancias, ya que rompe la relación, imprescindible, entre la intensidad de la señal y la posición en el vector Fingerprint del AP que se ha realizado en el diseño del sistema.

5.3 MEDIDAS DE DIFERENCIACIÓN

- El algoritmo k-NN presente en un entorno ruidoso, obtiene porcentajes más eficientes de localización con la distancia Canberra, ya que, según su fórmula, permite suavizar la diferencia entre potencias, fruto del multipath, que pueden influir negativamente en la decisión del algoritmo.

En el cuadro que se presenta a continuación, se recogen los experimentos que se han realizado, con su correspondiente porcentaje óptimo e indicando la distancia, número de vecinos y tamaño del conjunto de entrenamiento que se ha utilizado para ello.

Tabla 4: Resultados óptimos.

Experimento	Resultado	Distancia	Número de vecinos	Grupo de testing	Número de AP's
K y N variables sin k=1 y sin normalización	84,72%	Euclídea	k=3,4,6 y 7	40%	N=68
K y N variables sin k=1 y con normalización	84,72%	Euclídea	k=3,6	40%	N=68
K y N variables con k=1 y sin normalización	87,50%	Euclídea	k=4	40%	N=68
K y N variables con k=1 y con normalización	87,50%	Euclídea	k=2,3 y 4	40%	N=68
Sin k=1, sin normalizar y diferentes distancias, N=68	87,50%	Canberra	k=2,3 y 4	40%	N=68
Sin k=1, con normalizar y diferentes distancias, N=68	86,10%	Canberra	k=3,4 y 5	40%	N=68
Con k=1, sin normalizar y diferentes distancias, N=68	88,89%	Canberra	k=2,3 y 4	40%	N=68
Con k=1, con normalizar y diferentes distancias, N=68	88,89%	Canberra	k=4	40%	N=68

También se añade que el 40% de testing permite menor probabilidad de error. El resultado a destacar es de un 88,89%, que se obtiene con el número máximo de AP's disponibles en el emplazamiento, con un 60% de instancias de training, añadiendo la propiedad del peso (k=1) y usando la distancia Canberra. El número de vecinos que ayuda a obtener este porcentaje exitoso oscila entre k=2 a k=4.

5.4 MODIFICACIONES DEL PROCEDIMIENTO

5.4.1 K-dd dinámico

Permite que el proceso de evaluación sea más rápido, porque limita el número de instancias sobre las que se aplica la medida de similitud, sólo a aquellas que pertenecen a zonas contiguas de la nueva localización. Elimina la indeterminación de los resultados, ya que evita la presencia de muestras fuertes no contiguas a la zona donde el número de AP's comunes entre la nueva instancia y la de training sea mayor.

5.4.2 ENN

Eliminando las instancias ruidosas y/o producto del multipath del conjunto de entrenamiento, se alcanza el sistema de localización óptimo, llegando a un 97% de localizaciones exitosas. Se demuestra la influencia negativa que puede tener el multipath y desvanecimientos de señales de potencia en un entorno indoor.

5.5 TRABAJOS FUTUROS

A continuación, se proponen una serie mejoras que pueden optimizar y completar el proyecto realizado.

5.5.1 Integración de un sistema de localización completo

El sistema de localización que se ha estudiado, se puede completar con una interfaz interactiva que represente de forma gráfica y acústica, la posición de un nuevo usuario que se ha obtenido como resultado de la ejecución de la técnica Fingerprint y el algoritmo k-NN que se ha estudiado.

5.5.2 Optimización del código

La técnica Fingerprint y el algoritmo k-NN que componen el sistema de localización, se han implementado con un conjunto de clases Java (capítulo 3). El objetivo del código que las componen es que sea un código abierto, es decir, que sea la base para futuros desarrolladores que quieran aplicarle posibles mejoras, como por ejemplo, optimizar la velocidad de procesamiento.

5.5.3 k-NN aplicado a objetos en movimiento

El algoritmo k-NN [Gurram] [Kyriakos Mouratidis, 2000] [Yufei Tao, 2002] [Koudas] que se ha desarrollado, sólo contempla objetos estáticos. Una posible mejora sería aquella que permitiera obtener la siguiente posición de un usuario pasado un período de tiempo T.

La variación del algoritmo k-NN que introduce la búsqueda dinámica es CRNN (Continus Reverse Nearest Neighbour). Consiste en buscar las áreas más cercanas al objeto en movimiento en un determinado intervalo de tiempo. Cada resultado se presenta como un vector que va acompañado del intervalo $[t - t_0]$. Este intervalo hace referencia al período de tiempo en el que una instancia ha sido el vecino más cercano del nuevo usuario a localizar.

Uno de los posibles algoritmos que permite implementar esta función es R-trees. El espacio de localización se divide en diferentes regiones que son las que forman los nodos del árbol. La estructura se conoce como “grid space”. El método consiste en comenzar desde el origen del árbol y elegir la rama cuya mínima distancia sea inferior al vecino más cercano ya encontrado. Este proceso se repetiría recursivamente hasta alcanzar una hoja del árbol. Cada $T' < T$ el algoritmo recibirá una nueva entrada a calcular la nueva posición del usuario que tiene desde t_0 hasta t .

5.5.4 Localización en 3 Dimensiones

El entorno de localización descrito [Koudas], se ha realizado en una única planta del edificio Torres Quevedo, es decir, en un espacio de dos dimensiones. El algoritmo se podría extender a una dimensión más, donde la posición de un nuevo objeto quedaría determinado no sólo por x e y , sino, también por una nueva coordenada que indicara la altura a nivel de suelo, es decir, la planta en la que se encuentra.

ANEXO

FASE I

K y N variables

Tabla 5: Estudio del comportamiento del algoritmo variando los parámetros k y N

	N=3	N=5	N=7	N=36	N=68
k=2	38,89%	44%	38,89%	69,40%	77,78%
k=3	44%	44%	41,67%	66,70%	80,60%
k=4	41,67%	44%	41,67%	66,70%	72,20%
k=5	47,20%	44%	41,67%	69,40%	77,78%
k=6	44%	44%	36,10%	72,20%	72,20%
k=7	47,20%	47,20%	47,20%	72,20%	72,20%
k=8	44%	44%	38,89%	75%	69,40%

Pesos

Tabla 6: Comparativa del comportamiento del algoritmo sin la propiedad del peso y con ella para k $\in [2,8]$

	N=3	N=5	N=7	N=36	N=68
k=2	38,89%	44%	38,89%	69,40%	77,78%
k=2 con pesos	47,42%	50,00%	47,20%	72,20%	77,78%
k=3	44,00%	44%	41,67%	66,70%	80,60%
k=3 con pesos	44,00%	44,40%	44,00%	66,67%	80,56%
k=4	41,67%	44%	41,67%	66,70%	72,20%
k=4 con pesos	47,42%	44,40%	41,67%	69,40%	80,56%
k=5	47,20%	44%	41,67%	69,40%	77,78%
k=5 con pesos	47,42%	44,40%	41,67%	69,40%	77,78%
k=6	44,00%	44%	36,10%	72,20%	72,20%
k=6 con pesos	47,42%	47,20%	41,67%	77,78%	77,78%
k=7	47,20%	47%	47,20%	72,20%	72,20%
k=7 con pesos	52,78%	55,56%	50,00%	72,20%	75,00%
k=8	44,00%	44%	38,89%	75,00%	69,40%
k=8 con pesos	52,78%	52,78%	44,40%	80,56%	77,78%

Distancias

Tabla 7: Evaluación de diferentes distancias sin peso, sin normalizar y N=68

	Euclídea	City Block	Chevichef	Canberra
k=2	77,78%	80,56%	55,56%	80,56%
k=3	80,60%	80,56%	58,33%	80,56%
k=4	72,20%	75,00%	52,78%	80,56%
k=5	77,78%	77,78%	58,33%	80,56%
k=6	72,20%	72,20%	52,78%	72,20%
k=7	72,20%	72,20%	58,33%	77,78%
k=8	69,40%	69,40%	50,00%	72,20%

Tabla 8: Comparativa entre diferentes distancias con la propiedad el peso y sin normalizar

	Euclídea	City Block	Chevichef	Canberra
k=2	77,78%	80,56%	61,10%	80,56%
k=3	80,56%	80,56%	58,30%	80,56%
k=4	80,56%	83,30%	58,30%	80,56%
k=5	77,78%	77,78%	58,30%	83,30%
k=6	77,78%	77,78%	55,56%	75,00%
k=7	75,00%	75,00%	61,10%	77,78%
k=8	77,78%	77,78%	58,30%	72,20%

Normalización

Tabla 9: Comparativa del comportamiento del algoritmo sin y con normalización de datos para $k \in [2,8]$

	N=3	N=5	N=7	N=36	N=68
k=2	38,89%	44%	38,89%	69,40%	77,78%
k=2 con normalización	36,10%	44,40%	36,10%	66,70%	77,78%
k=3	44,00%	44%	41,67%	66,70%	80,60%
k=3 con normalización	47,20%	50,00%	47,20%	63,90%	77,78%
k=4	41,67%	44%	41,67%	66,70%	72,20%
k=4 con normalización	50,00%	52,80%	47,20%	63,90%	72,20%
k=5	47,20%	44%	41,67%	69,40%	77,78%
k=5 con normalización	50,00%	52,70%	44,40%	69,40%	75,00%
k=6	44,00%	44%	36,10%	72,20%	72,20%
k=6 con normalización	50,00%	50,00%	44,40%	72,20%	75,00%
k=7	47,20%	47%	47,20%	72,20%	72,20%
k=7 con normalización	61,10%	61,10%	61,10%	72,20%	72,20%
k=8	44,00%	44%	38,89%	75,00%	69,40%
k=8 con normalización	61,10%	61,10%	61,10%	72,20%	72,20%

Tabla 10: Comportamiento del algoritmo con la propiedad del peso y con los datos normalizados

	N=3	N=5	N=7	N=36	N=68
k=2	50,00%	58%	55,60%	69,40%	80,56%
k=3	53%	53%	50,00%	63,89%	77,78%
k=4	58,30%	61%	58,30%	66,67%	80,56%
k=5	52,78%	58%	47,20%	69,40%	77,78%
k=6	50%	53%	52,78%	72,20%	77,78%
k=7	63,89%	63,89%	66,67%	72,20%	72,20%
k=8	61%	61%	69,40%	75%	77,78%

Tabla 11: Comparación del comportamiento del algoritmo usando la propiedad del peso y con y sin normalización para $k \in [2, 8]$

	N=3	N=5	N=7	N=36	N=68
k=2 con normalización	50,00%	58%	55,60%	69,40%	80,56%
k=2	47,42%	50,00%	47,20%	72,20%	77,78%
k=3 con normalización	52,78%	53%	50,00%	63,89%	77,78%
k=3	44,00%	44,40%	44,00%	66,67%	80,56%
k=4 con normalización	58,30%	61%	58,30%	66,67%	80,56%
k=4	47,42%	44,40%	41,67%	69,40%	80,56%
k=5 con normalización	52,78%	58%	47,20%	69,40%	77,78%
k=5	47,42%	44,40%	41,67%	69,40%	77,78%
k=6 con normalización	50,00%	53%	52,78%	72,20%	77,78%
k=6	47,42%	47,20%	41,67%	77,78%	77,78%
k=7 con normalización	63,89%	64%	66,67%	72,20%	72,20%
k=7	52,78%	55,56%	50,00%	72,20%	75,00%
k=8 con normalización	61,10%	61%	69,40%	75,00%	77,78%
k=8	52,78%	52,78%	44,40%	80,56%	77,78%

Tabla 12: Comparativa de la importancia de la presencia del peso en el caso de los datos sin normalizar y normaliza

	N=3	N=5	N=7	N=36	N=68
k=2 sin peso	36,10%	44,40%	36,10%	66,70%	77,78%
k=2 con peso	50,00%	58,30%	55,60%	69,40%	80,56%
con normalización	-13,90%	-13,90%	-19,50%	-2,70%	-2,78%
k=2 sin peso	38,89%	44,00%	38,89%	69,40%	77,78%
k=2 con pesos	47,42%	50,00%	47,20%	72,20%	77,78%
sin normalizar	-8,53%	-6,00%	-8,31%	-2,80%	0,00%
	N=3	N=5	N=7	N=36	N=68
k=3 sin peso	47,20%	50%	47,20%	63,90%	77,78%
k=3 con peso	52,78%	52,78%	50,00%	63,89%	77,78%
con normalización	-5,58%	-2,78%	-2,80%	0,01%	0,00%
k=3 sin peso	44,00%	44,00%	41,67%	66,70%	80,60%
k=3 con pesos	44,00%	44,40%	44,00%	66,67%	80,56%
sin normalizar	0,00%	0,00%	-2,33%	0,03%	0,04%
	N=3	N=5	N=7	N=36	N=68
k=4 sin peso	50,00%	53%	47,20%	63,90%	72,20%
k=4 con peso	58,30%	61,10%	58,30%	66,67%	80,56%
con normalización	-8,30%	-8,30%	-11,10%	-2,77%	-8,36%
k=4	41,67%	44,00%	41,67%	66,70%	72,20%
k=4 con pesos	47,42%	44,40%	41,67%	69,40%	80,56%
sin normalizar	5,75%	0,40%	0,00%	2,70%	8,36%
	N=3	N=5	N=7	N=36	N=68
k=5 sin peso	50%	53%	44,40%	69,40%	75,00%
k=5 con peso	52,78%	58,30%	47,20%	69,40%	77,78%
con normalización	-2,78%	-5,60%	-2,80%	0,00%	-2,78%
k=5	47,20%	44,00%	41,67%	69,40%	77,78%
k=5 con pesos	47,42%	44,40%	41,67%	69,40%	77,78%
sin normalizar	0,22%	0,40%	0,00%	0,00%	0,00%
	N=3	N=5	N=7	N=36	N=68
k=6 sin peso	50%	50%	44,40%	72,20%	75,00%
k=6 con peso	50,00%	52,78%	52,78%	72,20%	77,78%
con normalización	0,00%	-2,78%	-8,38%	0,00%	-2,78%
k=6 sin peso	44,00%	44,00%	36,10%	72,20%	72,20%
k=6 con pesos	47,42%	47,20%	41,67%	77,78%	77,78%
sin normalizar	3,42%	3,20%	5,57%	5,57%	5,58%
	N=3	N=5	N=7	N=36	N=68
k=7 sin peso	61,10%	61,10%	61,10%	72,20%	72,20%
k=7 con peso	63,89%	63,89%	66,67%	72,20%	72,20%
con normalización	-2,79%	-2,79%	-5,57%	0,00%	0,00%
k=7 sin peso	47,20%	47,00%	47,20%	72,20%	72,20%
k=7 con pesos	52,78%	55,56%	50,00%	72,20%	75,00%
sin normalizar	5,58%	8,36%	2,80%	0,00%	2,80%

k=8 sin peso	61,10%	61,10%	61,10%	72,20%	72,20%
k=8 con peso	61,10%	61,10%	69,40%	75,00%	77,78%
con normalización	0,00%	0,00%	-8,30%	-2,80%	-5,58%
k=8 sin peso	44,00%	44,00%	38,89%	75,00%	69,40%
k=8 con pesos	52,78%	52,78%	44,40%	80,56%	77,78%
sin normalizar	8,78%	8,78%	5,51%	5,56%	8,38%

Tabla 13: Comparativa entre las diferentes distancias sin peso y con normalización

	Euclídea	City Block	Chevichef	Canberra
k=2	77,78%	80,56%	52,78%	80,56%
k=3	77,78%	77,78%	55,56%	80,56%
k=4	72,20%	72,20%	55,56%	72,20%
k=5	75,00%	75,00%	55,56%	72,20%
k=6	75,00%	75,00%	52,78%	72,20%
k=7	72,20%	75,00%	58,30%	72,20%
k=8	72,20%	75,00%	63,89%	72,20%

Tabla 14: Comparativa entre las diferentes distancias sin la propiedad del peso, normalizando y sin normalizar los datos

	Euclídea	City Block	Chevichef	Canberra
k=2 con normalización	77,78%	81%	52,78%	80,56%
k=2	77,78%	80,56%	55,56%	80,56%
k=3 con normalización	77,78%	78%	55,56%	80,56%
k=3	80,60%	80,56%	58,33%	80,56%
k=4 con normalización	72,20%	72%	55,56%	72,20%
k=4	72,20%	75,00%	52,78%	80,56%
k=5 con normalización	75,00%	75%	55,56%	72,20%
k=5	77,78%	77,78%	58,33%	80,56%
k=6 con normalización	75,00%	75%	52,78%	72,20%
k=6	72,20%	72,20%	52,78%	72,20%
k=7 con normalización	72,20%	75%	58,30%	72,20%
k=7	72,20%	72,20%	58,33%	77,78%
k=8 con normalización	72,20%	75%	63,89%	72,20%
k=8	69,40%	69,40%	50,00%	72,20%

Tabla 15: Comparativa entre diferentes distancias con la propiedad del peso y normalizando

	Euclídea	City Block	Chevichef	Canberra
k=2	80,56%	83,30%	58,30%	83,30%
k=3	77,78%	77,78%	58,30%	80,56%
k=4	80,56%	80,56%	55,56%	75,00%
k=5	77,78%	77,78%	55,56%	75,00%
k=6	77,78%	77,78%	52,78%	75,00%
k=7	72,20%	75,00%	58,30%	72,20%
k=8	77,78%	80,56%	63,89%	75,00%

Tabla 16: Comparativa entre las diferentes distancias normalizadas con y sin peso para $k \in [2,8]$

	Euclídea	City Block	Chevichef	Canberra
k=2 con peso	80,56%	83%	58,30%	83,30%
k=2 sin peso	77,78%	80,56%	52,78%	80,56%
k=3 con peso	77,78%	78%	58,30%	80,56%
k=3 sin peso	77,78%	77,78%	55,56%	80,56%
k=4 con peso	80,56%	81%	55,56%	75,00%
k=4 sin peso	72,20%	72,20%	55,56%	72,20%
k=5 con peso	77,78%	78%	55,56%	75,00%
k=5 sin peso	75,00%	75,00%	55,56%	72,20%
k=6 con peso	77,78%	78%	52,78%	75,00%
k=6 sin peso	75,00%	75%	52,78%	72,20%
k=7 con peso	72,20%	75%	58,30%	72,20%
k=7 sin peso	72,20%	75%	58,30%	72,20%
k=8 con peso	77,78%	81%	63,89%	75,00%
k=8	72,20%	75%	63,89%	72,20%

Tabla 17: Comparativa entre las diferentes distancias con la propiedad del peso, normalizando y sin normalizar para $k \in [2,8]$

	Euclídea	City Block	Chevichef	Canberra
k=2 con norma	80,56%	83%	58,30%	83,30%
k=2	77,78%	80,56%	61,10%	80,56%
k=3 con norma	77,78%	78%	58,30%	80,56%
k=3	80,56%	80,56%	58,30%	80,56%
k=4 con norma	80,56%	81%	55,56%	75,00%
k=4	80,56%	83,30%	58,30%	80,56%
k=5 con norma	77,78%	78%	55,56%	75,00%
k=5	77,78%	77,78%	58,30%	83,30%
k=6 con norma	77,78%	78%	52,78%	75,00%
k=6	77,78%	77,78%	55,56%	75,00%
k=7 con norma	72,20%	75%	58,30%	72,20%
k=7	75,00%	75,00%	61,10%	77,78%
k=8 con norma	77,78%	81%	63,89%	75,00%
k=8	77,78%	77,78%	58,30%	72,20%

Estudio para las diferentes distancias

Tabla 18: Resultados obtenidos con la Distancia Euclídea

	Sin k=1, Sin norma	Sin k=1, Con norma	Con k=1, Sin normal	Con k=1 y Con norma
k=2	77,78%	77,78%	77,78%	80,56%
k=3	80,56%	77,78%	80,56%	77,78%
k=4	72,20%	72,20%	80,56%	80,56%
k=5	77,78%	75,00%	77,78%	77,78%
k=6	72,20%	75,00%	77,78%	77,78%
k=7	72,20%	72,20%	75,00%	72,20%
k=8	69,40%	72,20%	77,78%	77,78%

Tabla 19: Resultados obtenidos con la distancia City Block

	sin k=1, sin norma	sin k=1, con norma	con k=1, sin normal	con k=1 y con norma
k=2	80,56%	80,56%	80,56%	83,30%
k=3	80,56%	77,78%	80,56%	77,78%
k=4	75,00%	72,20%	83,30%	80,56%
k=5	77,78%	75,00%	77,78%	77,78%
k=6	72,20%	75,00%	77,78%	77,78%
k=7	72,20%	75,00%	75,00%	75,00%
k=8	69,40%	75,00%	77,78%	80,56%

Tabla 20: Resultados obtenidos con la distancia Chebichef.

	sin k=1, sin norma	sin k=1, con norma	con k=1, sin normal	con k=1 y con norma
k=2	55,56%	52,78%	61,10%	58,30%
k=3	58,33%	55,56%	58,30%	58,30%
k=4	52,78%	55,56%	58,30%	55,56%
k=5	58,33%	55,56%	58,30%	55,56%
k=6	52,78%	52,78%	55,56%	52,78%
k=7	58,33%	58,30%	61,10%	58,30%
k=8	50,00%	63,89%	58,30%	63,89%

Tabla 21: Resultados obtenidos con la distancia Canberra

	sin k=1, sin norma	sin k=1, con norma	con k=1, sin normal	con k=1 y con norma
k=2	80,56%	80,56%	80,56%	83,30%
k=3	80,56%	80,56%	80,56%	80,56%
k=4	80,56%	72,20%	80,56%	75,00%
k=5	80,56%	72,20%	83,30%	75,00%
k=6	72,20%	72,20%	75,00%	75,00%
k=7	77,78%	72,20%	77,78%	72,20%
k=8	72,20%	72,20%	72,20%	75,00%

FASE II: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento

Tabla 22: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, sin peso y sin normalización para $k \in [2, 7]$

	N=3	N=5	N=7	N=36	N=68
k=2 20%	38,89%	44%	38,89%	69,40%	77,78%
k=2 30%	40,74%	48,15%	42,60%	72,20%	83,30%
k=2 40%	38,90%	43,05%	36,10%	73,60%	81,94%
k=3 20%	44,00%	44%	41,67%	66,70%	80,60%
k=3 30%	46,30%	48,00%	48,00%	77,80%	83,30%
k=3 40%	43,06%	44,40%	52,70%	77,80%	84,72%
k=4 20%	41,67%	44%	41,67%	66,70%	72,20%
k=4 30%	40,74%	46,30%	55,60%	79,60%	77,78%
k=4 40%	43,06%	44,40%	47,20%	79,17%	84,72%
k=5 20%	47,20%	44%	41,67%	69,40%	77,78%
k=5 30%	48,00%	46,30%	53,70%	81,48%	81,50%
k=5 40%	51,39%	54,10%	52,70%	76,39%	83,30%
k=6 20%	44,00%	44%	36,10%	72,20%	72,20%
k=6 40%	50,00%	52,78%	48,61%	76,39%	84,72%
k=6 30%	50,00%	51,85%	50,00%	75,90%	79,63%
k=7 20%	47,20%	47%	47,20%	72,20%	72,20%
k=7 30%	48,00%	51,85%	50,00%	77,80%	77,78%
k=7 40%	45,83%	48,60%	47,20%	77,80%	84,72%

Tabla 23: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de training, N variable, sin peso y con normalización para $k \in [2,7]$

	N=3	N=5	N=7	N=36	N=68
k=2 20%	36,10%	44%	36,10%	66,70%	77,78%
k=2 30%	44,40%	50,00%	46,30%	70,37%	81,48%
k=2 40%	45,83%	50,00%	50,00%	70,80%	80,57%
k=3 20%	47,20%	50%	47,20%	63,90%	77,78%
k=3 30%	55,60%	59,26%	57,40%	74,07%	81,48%
k=3 40%	58,30%	66,70%	58,30%	76,39%	84,70%
k=4 20%	50,00%	53%	47,20%	63,90%	72,20%
k=4 30%	50,00%	55,60%	59,26%	77,00%	77,78%
k=4 40%	52,78%	52,80%	55,60%	77,78%	84,70%
k=5 20%	50,00%	53%	44,40%	69,40%	75,00%
k=5 30%	53,70%	57,40%	53,70%	75,93%	79,63%
k=5 40%	52,78%	55,56%	56,94%	80,56%	83,30%
k=6 20%	50,00%	50%	44,40%	72,20%	75,00%
k=6 30%	51,85%	55,60%	57,40%	77,80%	79,63%
k=6 40%	55,60%	58,30%	59,72%	80,56%	84,70%
k=7 20%	61,10%	61%	61,10%	72,20%	72,20%
k=7 30%	55,56%	55,56%	62,96%	77,80%	79,63%
k=7 40%	58,30%	56,94%	61,10%	77,78%	83,30%

Tabla 24: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento, N variable, con peso y sin normalización para $k \in [2,7]$

	N=3	N=5	N=7	N=36	N=68
k=2 20%	47,42%	50%	47,20%	72,20%	77,78%
k=2 30%	51,85%	55,60%	53,70%	77,78%	85,19%
k=2 40%	48,10%	50,00%	55,60%	80,56%	87,50%
k=3 20%	44,00%	44%	44,00%	66,67%	80,56%
k=3 30%	50,00%	51,85%	53,70%	77,78%	83,30%
k=3 40%	45,83%	50,00%	56,94%	79,17%	87,50%
k=4 20%	47,42%	44%	41,67%	69,40%	80,56%
k=4 30%	51,85%	51,85%	57,40%	79,63%	85,19%
k=4 40%	52,80%	55,60%	52,78%	81,94%	87,50%
k=5 20%	47,42%	44%	41,67%	69,40%	77,78%
k=5 30%	55,56%	53,70%	55,56%	83,30%	81,48%
k=5 40%	51,39%	55,56%	54,17%	79,17%	83,30%
k=6 20%	47,42%	47%	41,67%	77,78%	77,78%
k=6 30%	53,70%	53,70%	53,70%	81,48%	85,19%
k=6 40%	51,39%	54,17%	52,78%	76,39%	86,10%
k=7 20%	52,78%	56%	50,00%	72,20%	75,00%
k=7 30%	51,85%	55,56%	50,00%	79,63%	79,63%
k=7 40%	50,00%	54,17%	48,61%	77,78%	86,10%

Tabla 25: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento, N variable, con peso y con normalización para $k \in [2,7]$

	N=3	N=5	N=7	N=36	N=68
k=2 20%	50,00%	58%	55,60%	69,40%	80,56%
k=2 30%	62,96%	66,70%	61,10%	75,93%	85,19%
k=2 40%	62,28%	69,40%	68,05%	77,78%	87,50%
k=3 20%	52,78%	53%	50,00%	63,89%	77,78%
k=3 30%	61,10%	62,96%	61,10%	74,07%	81,48%
k=3 40%	66,67%	73,61%	68,06%	77,78%	87,50%
k=4 20%	58,30%	61%	58,30%	66,67%	80,56%
k=4 30%	57,40%	62,96%	62,96%	81,48%	83,30%
k=4 40%	61,10%	66,67%	66,67%	79,17%	87,50%
k=5 20%	52,78%	58%	47,20%	69,40%	77,78%
k=5 30%	57,40%	61,10%	55,56%	75,93%	81,48%
k=5 40%	61,10%	63,89%	61,10%	80,56%	84,72%
k=6 20%	50,00%	53%	52,78%	72,20%	77,78%
k=6 30%	53,70%	59,26%	62,96%	79,63%	83,30%
k=6 40%	62,50%	59,26%	68,05%	80,56%	86,10%
k=7 20%	63,89%	64%	66,67%	72,20%	72,20%
k=7 30%	59,26%	61,10%	68,52%	77,78%	79,63%
k=7 40%	61,10%	62,28%	66,67%	79,17%	84,72%

Tabla 26: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training y distancias, sin peso y sin normalización para $k \in [2,8]$

	Euclídea	City Block	Chevichef	Canberra
k=2 20%	77,78%	81%	55,56%	80,56%
k=2 30%	83,30%	83,30%	59,26%	83,30%
k=2 40%	81,94%	81,90%	55,56%	87,50%
k=3 20%	80,60%	81%	58,33%	80,56%
k=3 30%	83,30%	83,30%	66,67%	85,19%
k=3 40%	84,72%	84,72%	66,67%	87,50%
k=4 20%	72,20%	75%	52,78%	80,56%
k=4 30%	77,78%	77,80%	59,26%	83,30%
k=4 40%	84,72%	83,30%	62,50%	87,50%
k=5 20%	77,78%	78%	58,33%	80,56%
k=5 30%	81,50%	83,30%	64,80%	81,48%
k=5 40%	83,30%	83,30%	65,28%	84,72%
k=6 20%	72,20%	72%	52,78%	72,20%
k=6 30%	79,63%	77,80%	62,96%	81,48%
k=6 40%	84,72%	84,72%	58,30%	86,10%
k=7 20%	72,20%	72%	58,33%	77,78%
k=7 30%	77,78%	77,80%	66,67%	81,48%
k=7 40%	84,72%	83,30%	61,10%	86,10%

Tabla 27: Comparativa del comportamiento del algoritmo con diferentes tamaños del grupo de entrenamiento y distancias, sin peso y con normalización para $k \in [2,7]$

	Euclídea	City Block	Chevichef	Canberra
k=2 20%	77,78%	81%	52,78%	80,56%
k=2 30%	81,48%	81,48%	59,26%	85,20%
k=2 40%	80,57%	80,56%	55,56%	84,72%
k=3 20%	77,78%	78%	55,56%	80,56%
k=3 30%	81,48%	81,48%	67,04%	83,30%
k=3 40%	84,70%	84,72%	70,83%	86,10%
k=4 20%	72,20%	72%	55,56%	72,20%
k=4 30%	77,78%	77,78%	70,37%	81,48%
k=4 40%	84,70%	84,72%	69,40%	86,10%
k=5 20%	75,00%	75%	55,56%	72,20%
k=5 30%	79,63%	79,63%	70,37%	85,20%
k=5 40%	83,30%	83,30%	72,20%	86,10%
k=6 20%	75,00%	75%	52,78%	72,20%
k=6 30%	79,63%	79,63%	68,52%	81,48%
k=6 40%	84,70%	83,30%	73,60%	83,10%
k=7 20%	72,20%	75%	58,30%	72,20%
k=7 30%	79,63%	79,63%	70,37%	81,48%
k=7 40%	83,30%	83,30%	75,00%	84,72%

Tabla 28: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de training y distancias, con peso y sin normalización para $k \in [2,7]$

	Euclídea	City Block	Chevichef	Canberra
k=2 20%	77,78%	81%	61,10%	80,56%
k=2 30%	85,19%	85,19%	70,37%	85,19%
k=2 40%	87,50%	87,50%	75,00%	88,89%
k=3 20%	80,56%	81%	58,30%	80,56%
k=3 30%	83,30%	83,30%	68,52%	85,19%
k=3 40%	87,50%	87,50%	69,40%	88,89%
k=4 20%	80,56%	83%	58,30%	80,56%
k=4 30%	85,19%	83,30%	66,67%	85,19%
k=4 40%	87,50%	87,50%	72,20%	88,89%
k=5 20%	77,78%	78%	58,30%	83,30%
k=5 30%	81,48%	83,30%	64,81%	83,30%
k=5 40%	83,30%	83,30%	70,83%	86,10%
k=6 20%	77,78%	78%	55,56%	75,00%
k=6 30%	85,19%	83,30%	66,67%	81,48%
k=6 40%	86,10%	84,72%	69,40%	86,10%
k=7 20%	75,00%	75%	61,10%	77,78%
k=7 30%	79,63%	79,63%	66,67%	81,48%
k=7 40%	86,10%	84,72%	66,67%	87,50%

Tabla 29: Comparativa del comportamiento del algoritmo con diferentes tamaños del conjunto de entrenamiento y distancias, con peso y con normalización para $k \in [2,7]$

	Euclídea	City Block	Chevichef	Canberra
k=2 20%	80,56%	83%	58,30%	83,30%
k=2 30%	85,19%	85,19%	66,67%	87,04%
k=2 40%	87,50%	87,50%	69,40%	87,50%
k=3 20%	77,78%	78%	58,30%	80,56%
k=3 30%	81,48%	81,48%	72,20%	83,30%
k=3 40%	87,50%	87,50%	73,61%	87,50%
k=4 20%	80,56%	81%	55,56%	75,00%
k=4 30%	83,30%	83,30%	70,37%	83,30%
k=4 40%	87,50%	87,50%	75,00%	88,89%
k=5 20%	77,78%	78%	55,56%	75,00%
k=5 30%	81,48%	81,48%	70,37%	87,04%
k=5 40%	84,72%	84,72%	75,00%	87,50%
k=6 20%	77,78%	78%	52,78%	75,00%
k=6 30%	83,30%	83,30%	72,20%	83,30%
k=6 40%	86,10%	84,72%	76,39%	86,10%
k=7 20%	72,20%	75%	58,30%	72,20%
k=7 30%	79,63%	79,63%	70,37%	81,48%
k=7 40%	84,72%	84,72%	79,17%	86,10%

Filtrado de ruido

Tabla 31: Comparativa entre KNN y ENN con k variable, sin peso y sin normalización de datos

	Euclídea	CityBlock	Chevichef	Canberra
k=2_ENN	97,00%	97%	91,17%	97,00%
k=2	77,78%	80,56%	55,56%	80,56%
k=3_ENN	97,00%	97%	91,17%	97,00%
k=3	80,60%	80,56%	58,33%	80,56%
k=4_ENN	97,00%	97%	70,59%	94,10%
k=4	72,20%	75,00%	52,78%	80,56%
k=5_ENN	97,00%	97%	82,30%	97,00%
k=5	77,78%	77,78%	58,33%	80,56%
k=6_ENN	94,10%	94%	73,50%	94,10%
k=6	72,20%	72,20%	52,78%	72,20%
k=7_ENN	94,10%	94%	79,40%	94,10%
k=7	72,20%	72,20%	58,33%	77,78%
k=8_ENN	94,10%	94%	67,65%	94,10%
k=8	69,40%	69,40%	50,00%	72,20%

Tabla 32: Comparativa entre KNN y ENN con k variable, con peso y sin normalización de datos

	Euclídea	CityBlock	Chevichef	Canberra
k=2_ENN	97,00%	97%	97,00%	97,00%
k=2	77,78%	80,56%	61,10%	80,56%
k=3_ENN	97,00%	97%	91,17%	97,00%
k=3	80,56%	80,56%	58,30%	80,56%
k=4_ENN	97,00%	97%	91,17%	97,00%
k=4	80,56%	83,30%	58,30%	80,56%
k=5_ENN	97,00%	97%	85,30%	97,00%
k=5	77,78%	77,78%	58,30%	83,30%
k=6_ENN	97,00%	97%	82,30%	97,00%
k=6	77,78%	77,78%	55,56%	75,00%
k=7_ENN	94,10%	94%	88,20%	94,10%
k=7	75,00%	75,00%	61,10%	77,78%
k=8_ENN	94,10%	94%	88,20%	94,10%
k=8	77,78%	77,78%	58,30%	72,20%

Tabla 33: Comparativa entre KNN y ENN con k variable, sin peso y con normalización de datos

	Euclídea	CityBlock	Chevichef	Canberra
k=2_ENN	97,00%	97%	85,30%	94,10%
k=2	77,78%	80,56%	52,78%	80,56%
k=3_ENN	97,00%	97%	88,20%	94,10%
k=3	77,78%	77,78%	55,56%	80,56%
k=4_ENN	97,00%	97%	85,30%	94,10%
k=4	72,20%	72,20%	55,56%	72,20%
k=5_ENN	97,00%	97%	82,35%	94,10%
k=5	75,00%	75,00%	55,56%	72,20%
k=6_ENN	97,00%	94%	76,47%	94,10%
k=6	75,00%	75,00%	52,78%	72,20%
k=7_ENN	94,10%	97%	82,30%	94,10%
k=7	72,20%	75,00%	58,30%	72,20%
k=8_ENN	94,10%	97%	79,40%	94,10%
k=8	72,20%	75,00%	63,89%	72,20%

Tabla 34: Comparativa entre KNN y ENN con k variable, con peso y con normalización de datos

	Euclídea	CityBlock	Chevichef	Canberra
k=2_ENN	97,00%	97%	97,00%	97,00%
k=2	80,56%	83,30%	58,30%	83,30%
k=3_ENN	97,00%	97%	91,17%	94,10%
k=3	77,78%	77,78%	58,30%	80,56%
k=4_ENN	97,00%	97%	94,10%	94,10%
k=4	80,56%	80,56%	55,56%	75,00%
k=5_ENN	97,00%	97%	85,30%	94,10%
k=5	77,78%	77,78%	55,56%	75,00%
k=6_ENN	97,00%	97%	91,70%	94,10%
k=6	77,78%	77,78%	52,78%	75,00%
k=7_ENN	94,10%	97%	85,30%	94,10%
k=7	72,20%	75,00%	58,30%	72,20%
k=8_ENN	94,10%	97%	88,20%	97,00%
k=8	77,78%	80,56%	63,89%	75,00%

BIBLIOGRAFÍA

- Ahmad Hatami, K. P. (s.f.). On RSS and TOA based Indoor Geolocation – A Comparative Performance Evaluation. Obtenido de http://users.wpi.edu/~mheidari/files/WCNC_hatami.pdf
- Barrett, T. W. (Julio de 2000). History of UltraWideBand (UWB) Radar&Communications: Pioneers and Innovators. Obtenido de [http://www.ntia.doc.gov/osmhome/owbtestplan/barret_history_\(piersw-figs\).pdf](http://www.ntia.doc.gov/osmhome/owbtestplan/barret_history_(piersw-figs).pdf)
- Cáceres, J. C. (2003). Redes Neuronales. Obtenido de http://www.usp.edu.pe/~jc.gutierrez/IA_IRN.pdf
- Caviedes, S. M. (1999). El estándar de comunicaciones por infrarrojos IrDA. Obtenido de <http://www.monografias.com/trabajos24/estandar-comunicaciones-irda/estandar-comunicaciones-irda.shtml>
- Chahé Nerguizian, Charles Despins and Sofiéne Affe., (2001). A framework for Indoor Geolocation using and Intelligent System. 3rd WLAN Workshop.
- ConocimientosWeb.net (2005). Obtenido de <http://www.conocimientosweb.net/zip/article1107.html>
- D. Randall Wilson, T. R. (2000). Reduction Techniques for Instance-Based. Obtenido de Neural Network & Machine Learning Laboratory, Computer Science Department, Brigham Young University,: <http://www.springerlink.com/content/n748p82037443824/fulltext.pdf>
- Fernandez, C. (2007). Aprendizaje Automático y Data Mining. Bloque III: Métodos de aprendizaje inductivo. Obtenido de http://isa.umh.es/ asignaturas/aprendizaje/3_metodos.pdf
- Fontal, B. (2006). El Espectro Electromagnético y sus. Obtenido de http://www.saber.ula.ve/bitstream/123456789/16746/1/espectro_electromagnetico.pdf
- Gruop, B. S. (2000). Obtenido de <http://spanish.bluetooth.com/>
- Gurram, r. p. (s.f.). A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects. Obtenido de <http://kirans.net/Documents/Data%20Base%20seminar-1.ppt>
- Hightower, J., & Borriello, G. (2001). Location Sensing Techniques. Obtenido de <http://seattle.intel-research.net/people/jhightower/pubs/hightower2001techniques/hightower2001techniques.pdf>
- Kim, J. H. (1999). A Fuzzy K-NN Algorithm using Weights from the Variance of Membership Values. Obtenido de <http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/6370/17024/00784711.pdf?arnumber=784711>
- Koudas, X. Y. (s.f.). Monitoring k-Nearest Neighbor Queries Over Moving Objects. Obtenido de http://www.yorku.ca/xhyu/publications/yux_knn.pdf
- Kyriakos Mouratidis, M. L. (2006). Continuous Nearest Neighbor Monitoring in Road Networks. Obtenido de <http://www.vldb.org/conf/2006/p43-mouratidis.pdf>
- Li, K.-H. (2000). Multipath. Obtenido de <http://www.ece.gatech.edu/research/labs/sarl/>
- López, D. G. (1999). Aplicación del muestreo Bayesiano en robots móviles: Estrategias para localización y estimación de mapas del entorno. Obtenido de <http://www.dccia.ua.es/ria/artics/tesis-domingo.pdf>
- M.F. Cátedra, J. G. (2007). Application of High Frequencies Techniques for Location Systems. Obtenido de http://www.ursi.org/B/EMTS_2007/O2-49/7-Catedra231.pdf
- Mauricio Henrique Costa Dias, A. G. (2005). Indoor TDOA-AOA Measurements at the 3G. Obtenido de http://iecom.dee.ufcg.edu.br/~jcis/dezembro2005/volume20/JCIS_2005_20_004_On.pdf

- Montero, J. M. (s.f.). Técnicas MBL de aprendizaje. Obtenido de <http://lorien.die.upm.es/~juancho/pfcs/JAA/capitulo2.pdf>
- Pahlavan, K. X. (2002). Indoor geolocation science and technology. Obtenido de http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=983917
- Patmanathan, V. (2006:008). Area Localization using WLAN. Obtenido de http://www.ee.kth.se/php/modules/publications/reports/2006/XR-EE-KT_2006_008.pdf
- Petru, A. M. (2006). Designing and performance evaluation for an indoor location and tracking system. Obtenido de <http://www.ijoe.org/ojs/include/getdoc.php?id=195&article=53&mode=pdf>
- Roberto Battiti, A. V. (2002). Neural network models for intelligent networks; Universit`a di Trento, Dipartimento di Informatica e Telecomunicazioni. Obtenido de <http://www.cains.cs.ucla.edu/ains2002/resource/battiti.pdf>
- Ruiz, D. A. (CEDI 2005). Tecnologías sensoriales de localización para entornos inteligentes. . Obtenido de http://www.proteinlab.cl/arhiv_transcape/Tecnologias%20localizacion%20entornos%20inteligentes.pdf
- Sinan Gezici, Z. T. (2005). Localization via Ultra-Wideband Radios. A look at positioning aspects of future sensor networks. IEEE Signal Processing Magazine. Obtenido de http://ens.ewi.tudelft.nl/Education/courses/et4248/uwblocalization_gezici.pdf
- Systems, T. S.-C. (2005). Multipath and Diversity. Obtenido de <http://www.cisco.com/application/pdf/paws/27147/multipath.pdf>
- Systems., M. (Agosto de 2005). RFID: Introducción a la Tecnología. Obtenido de <http://www.mobile.com.ar/pdfs/General.RFIDIntroducción.WhitePaper01.20050820M.pdf>
- Teknomo, K. (2006). Obtenido de <http://people.revoledu.com/kardi/tutorial/KNN/index.html>
- Téllez, L. M. (s.f.). Infrasonidos y Ultrasonidos. Obtenido de http://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_03_04/infra_y_ultra/efectos_ultrasonidos.htm
- Ting Liu, A. W. (1999). An Investigation of Practical Approximate Nearest Neighbor Algorithms. Obtenido de <http://www.cs.cmu.edu/~agray/approxnn.pdf>
- TOSHIBA. (s.f.). RFID - Spanish Information. Obtenido de <http://www.toshiba-tro.de/rfid/pages/spanish.html>
- Tutorial-Report.com. (2005:07). Wireless LAN (Wifi) Tutorial. Obtenido de http://www.tutorial-reports.com/wireless/wlanwifi/introduction_wifi.php
- Urraza, J. d., & YAEL, U. (2007). Identificación por radiofrecuencia. Obtenido de <http://www.jeuazarru.com/docs/RFID.pdf>
- Uzair Ahmad1, A. G. (2006). In-building Localization using Neural Networks; Department of Computer Engineering, Kyung Hee University, Korea1. Obtenido de <http://uclab.khu.ac.kr/ar/~uzair/pdfs/C9.pdf>
- Widyawan, M. K. (s.f.). Influence of Predicted and Measured; Centre for Adaptive Wireless Systems. Obtenido de <http://www.aws.cit.ie/personnel/Papers/Paper285.pdf>
- Wikipeda. (s.f.). Triangulación. Obtenido de <http://en.wikipedia.org/wiki/Triangulation>
- Wikipedia. (s.f.). Obtenido de <http://es.wikipedia.org/wiki/WLAN>
- Wikipedia. (2007). Obtenido de <http://en.wikipedia.org/wiki/Ultra-wideband>
- Wikipedia. (1994). Bluetooth. Obtenido de <http://es.wikipedia.org/wiki/Bluetooth#Historia>

Wilson, J. M. (2002). Development Ultra-Wideband/a Disruptive RF Technology? Obtenido de http://www.intel.com/technology/comms/uwb/download/Ultra-Wideband_Technology.pdf

Yamasaki, R. O. (2005). TDOA location system for IEEE 802.11b WLAN. Obtenido de http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1424880

Yufei Tao, D. P. (2002). Continuous Nearest Neighbor Search. Obtenido de <http://citeseerx.ist.psu.edu/showciting;jsessionid=6B0C407195AA319FE9BD8E5EAC996DA1?cid=5142513>